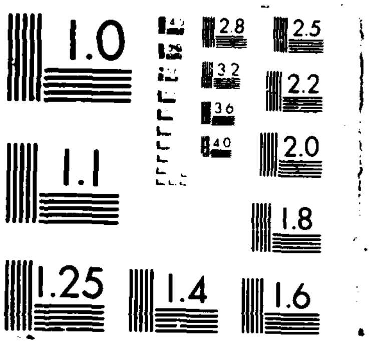
AD-A189 542 IMPLEMENTATION OF A TARGET STATE ESTIMATOR FOR THE 1/2
AIR-TO-AIR ATTACK MODE..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D L MICHALK
UNCLASSIFIED DEC 87 AFIT/GE/ENG/87D-44 F/G 19/5 NL


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

IMPLEMENTATION OF A TARGET STATE ESTIMATOR FOR THE AIR-TO-AIR ATTACK MODE OF THE AFTI/F-16

THESIS

David L. Michalk
Captain, USAF

AFIT/GE/EE/83D-44

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
[illegible]
[illegible]

[illegible]

①

AFIT/GE/ENG/87D-44

IMPLEMENTATION OF A TARGET STATE ESTIMATOR FOR THE AIR-TO-AIR ATTACK MODE OF THE AFTI/F-16

THESIS

David L. Michalk
Captain, USAF

AFIT/GE/ENG/87D-44

DTIC
ELECTE
MAR 0 2 1988
S D
H

Approved for public release; distribution unlimited

AFIT/GE/ENG/87D-44

IMPLEMENTATION OF A TARGET STATE ESTIMATOR FOR THE AIR-TO-AIR ATTACK MODE OF THE AFTI/F-16

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

David L. Michalk, B.S.E.E
Captain, USAF

December 1987

Approved for Public Release; Distribution Unlimited.

Preface

This study focused on the target state estimator, implemented as an extended Kalman filter, on the AFTI/F-16. Several possible reasons for the poor performance it exhibits are investigated; the major reason for this poor performance is shown to lie chiefly with the conventional covariance update techniques it uses. Others have shown that the recursion equations which use conventional update techniques are numerically unstable; because alternatives exist, the conventional updates should never be used.

I have had a lot of help from others in the implementation and writing of this thesis. I owe a large debt to my thesis advisor, Major W. H. Worsley, for his help in developing the theoretical aspects of this work as well as in the guidance provided in analyzing problems and in writing this document. Many thanks are also due to Mr. Finley Barfield of the AFTI/F-16 office for providing the data necessary to simulate the target state estimator on the AFTI/F-16. Finally, I wish to thank my wife Toni for tolerating my late-night computer sessions while this thesis was being developed.

David L. Michalk

ion For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

DTIC
COPY
INSPECTED
1

Table of Contents

Preface ii

List of Figures v

List of Tables. vi

Abstract. vii

I. Introduction 1

  1.1. Background 1
  1.2. Problem Statement. 3
  1.3. Scope and Limitations. 4
  1.4. Equipment Used 6
  1.5. Organization 7

II. Review of Current Knowledge. 8

  2.1. Introduction 8
  2.1. AFTI/F-16 Aircraft 8
    2.2.1. Sensors 10
    2.2.2. Digital Flight Control System . . .12
  2.3. Integrated Fire/Flight Control Concepts.12
  2.4. Kalman Filtering 14
    2.4.1. Extended Kalman Filter Equations. .15
    2.4.2. Wordlength Problems 17
    2.4.3. U-D Factorization 18
    2.4.4. Reference Frames: Measurement vs. State 22
    2.4.5. Filter Implementation 24
  2.5. Summary. 25

III. Constant Turn Rate Filter. 26

  3.1. Introduction 26
  3.2. Filter Dynamics Model. 26
  3.3. Measurement Equations. 31
    3.3.1. Range Measurement 31
    3.3.2. Velocity Measurement. 32
    3.3.3. Angle Measurements. 33
    3.3.4. The Measurement Matrix. 36
  3.4. Noise Models 38
    3.4.1. Dynamics Noises 38
    3.4.2. Measurement Noises. 39
  3.5. CTR filter Implementation. 41
  3.6. Coordinate Transformations 45
  3.6. Summary. 46

IV. Gauss Markov Filter. 48

4.1. Introduction 48
4.2. Filter Dynamic Model 48
4.3. Measurement Equations. 50
4.4. Noise Models 52
4.5. Acceleration Rotation. 52
4.6. GM Filter Implementation 53
4.7. Summary. 59

V. Results and Conclusions. 60
5.1. Introduction 60
5.2. Trajectory Generation. 60
5.3. Filter Results 63
5.4. Conclusions. 64
5.5. Recommendations. 65

Appendix A. Euler Angle Development. 66

Appendix B. Constant Turn Rate Coefficient 69

Appendix C. Results of SOFE Runs 72

Bibliography. 127

Vita. 128

List of Figures

Figure Page

1. AFTI/F-16 Configuration 9
2. Digital Flight Control System 13
3. Covariances in Two Reference Frames 23
4. Measurement Angles. 34
5. Angular Error Covariances 50
6. Trajectory 1 Initial Conditions 62
7. Trajectory 2 Initial Conditions 62
8. CTR Filter, Trajectory 1. 73
9. GM Filter, Trajectory 1 82
10. GM Filter, U-D Implementation, Trajectory 1 . .91
11. CTR Filter, Trajectory 2. 100
12. GM Filter, Trajectory 2 109
13. GM Filter, U-D Implementation, Trajectory 2 . 118

List of Tables

Table Page

I. Measurement Accuracies and Variances. 40

II. Filter Results. 64

Abstract

The purpose of this study was to investigate target state estimation techniques for the air-to-air mode of the AFTI/F-16 automated maneuvering and attack system. The target state estimator (TSE) previously developed would not perform to specifications; possible reasons for this poor performance are presented as well as suggestions to upgrade the performance.

The TSE exists as extended Kalman filter equations in a digital computer. The previously developed Kalman filter equations used conventional covariance update techniques and a Gauss-Markov system dynamics model which expressed the states in an inertial reference frame. Measurements were performed in the line-of-sight (LOS) frame, but the covariance matrix was not rotated into the LOS frame during update. This study focused on three areas: (1) Determine if the Gauss-Markov dynamics model was adequate for the tracking accuracies specified. (2) Determine if a rotation had to be performed to account for the states being expressed in one frame while the measurements were physically made in another. (3) Determine what effect the conventional covariance updates, coupled with the short (16-bit) wordlength of the TSE computers, has on the stability of the Kalman filter.

Two filter dynamics models were designed, tested, and compared. The first model used complex equations and closely modeled an air-to-air engagement. Most of the complexity of

the model was maintained in its implementation, and it was used as a baseline model. The second filter used a Gauss-Markov dynamics model and made several assumptions to simplify computations.

Analysis of filter performance revealed that the Gauss-Markov filter dynamics model was, indeed, an adequate model. Also, the covariance matrix does not have to be rotated into the LOS frame if the measurements are redefined. The second filter was then implemented using U-D covariance factorization algorithms, but the time propagation routines used were apparently flawed. However, the poor performance of the TSE is no doubt caused by the conventional Kalman filter recursions, as they are inherently unstable.

IMPLEMENTATION OF A TARGET STATE ESTIMATOR FOR THE AIR-TO-AIR ATTACK MODE OF THE AFTI/F-16

I. Introduction

1.1. Background

The Flight Dynamics Laboratory (AFWAL/FIGX) is evaluating software for the Automated Maneuvering and Attack System (AMAS) for the Advanced Fighter Technology Integration AFTI/F-16 aircraft. The AMAS uses integrated fire/flight control (IFFC) techniques to track a designated target, calculate a flight condition from which a released weapon has a high kill probability, maneuver the aircraft to that flight condition, and release the weapon [2].

The AFTI/F-16 is capable of attacking ground as well as airborne targets; the AMAS should automate an attack against either. Because target maneuverability has a large affect on the tracking system's capabilities, the AMAS is subdivided into three modes: a stationary ground target mode, for use against structures or other non-mobile targets; a mobile ground target mode, for use against moving vehicles; and an airborne target mode, capable for use in an air-to-air engagement. The ground modes of the AMAS are currently implemented in the AFTI/F-16. However, the airborne mode is still under evaluation, partly because the AMAS tracking function does not always perform with the precision required to obtain a gun firing solution [2,10].

The AMAS tracking function consists of sensors, which detect the relative position and velocity of the target, and a target state estimator (TSE), which estimates current target states. The sensors are a forward-looking infrared (FLIR) imaging system and the inertial navigation unit (INU) for angle measurements, a laser range finder, and the APG-66 radar for range rate (Doppler) information. The TSE consists of an extended Kalman filter (EKF) algorithm implemented in a digital computer. The EKF calculates optimal (as described below) estimates of current target states [10].

The Fire Control Computer (FCC) performs the AMAS function of calculating the desired flight profile. A discussion of the algorithms used by the FCC to compute the desired flight profile is beyond the scope of this thesis. However, the knowledge that the FCC must be able to predict the target position and velocity at a specific time in the future implies that the target states must include position, velocity, and acceleration. To ensure that body rates do not influence the target states, the target states are expressed in an inertial Cartesian coordinate system. Therefore, the states estimated by the TSE are relative (target with respect to attacker) position, relative velocity, and total target acceleration expressed in an inertial Cartesian coordinate system [10].

The TSE uses an EKF to combine the measurements available from the sensors with a target dynamics model to generate the state estimates given above. These state estimates are the best estimates possible if the model used in the filter adequately represents the real world system and the variances used in the filter are truly representative of the variances in the real world system [4:4].

Theoretically, linear Kalman filters provide the best state estimates possible if the conditions described above are met; however, the algorithms for their implementation may not be numerically stable. This numerical instability is also apparent for some implementations of EKFs. In fact, the finite wordlength of digital computers may make the recursion equations of the Kalman filter diverge, as they do for certain conditions of the currently implemented AFTI/F-16 TSE [2]. The divergent effects of finite wordlength may be reduced by increasing the wordlength of the computer or by modifying the recursion equations which comprise the Kalman filter [4:368]. Computer wordlengths are difficult to change, whereas the recursion equations in software are readily modified. Possible modifications to the recursion algorithms are discussed further in Chapter II.

1.2. Problem Statement

The EKF currently implemented in the air-to-air mode of the AFTI/F-16 TSE diverges for certain attack geometries, causing the AMAS to break off the attack [2]. The specific geometries where the divergence occurs were not available to the researcher, but two possible causes for this divergence are addressed by this research. First, the current EKF uses standard covariance matrix update equations, which have been shown to be numerically unstable [1:338]. A modification to the EKF recursion algorithms is proposed by this thesis to eliminate, or at least significantly reduce, this problem. Second, the measurement covariances are known only in the line-of-sight (LOS) frame while the target states are known in the inertial frame. Consequently, a change of reference frames must be accomplished before the states and covariances are updated, but this is not done in the current TSE [10]. The rotation matrix to accomplish this change of reference frame and its use are discussed further in Chapter II.

1.3. Scope and Limitations

This thesis develops an EKF based upon the model used in the current AFTI/F-16 TSE and compares it to an EKF based upon a more complex model. The filters developed herein use numerically stable update algorithms, reducing or eliminating divergence due to the digital computer's finite wordlength. Several simplifying assumptions are made which significantly reduce the complexity of the development, as discussed below.

The first assumption is that the measurement coordinate frame (the LOS frame) is assumed to be inertially space-stabilized during the measurement. While this is not generally true of AFTI/F-16 hardware [10] (the sensor head rotates with the relative target position, as explained in Chapter II), any rotation is assumed to have second or higher order effects on TSE performance. This assumption eliminates the requirement to calculate Coriolis acceleration effects on the measurement; instead, errors introduced by Coriolis accelerations are assumed to be negligible compared to the dynamics noise models of the EKFs.

The next assumption is that the attacker's position, velocity, acceleration, and attitude are available from an inertial navigation unit (INU) without error. This assumption is made because current INUs have errors much smaller than the errors expected from the TSE [8:4]. Also, because any INU errors are also present in fire control computer calculations, the overall effect of INU errors on the AMAS solution is reduced [10].

Another assumption is that the parallax errors which occur because of the physical separation of the STS and the INU are negligible compared to other system errors. This is justifiable because the separations are small (less than 10 feet) [10].

A further assumption is that the earth-fixed (inertial) reference frame has its positive axes oriented in the local north, east, and down (towards the center of the earth) directions. The origin of this coordinate frame is fixed at the center of gravity (cg) of the attacker at the instant target designation takes place (when the TSE is at time zero).

The final major assumption is that the attacker is benign; i.e., the attacking aircraft does not maneuver. This assumption is made to simplify the implementation of testing routines. Formulating a complete model of the AFTI/F-16 control system and aerodynamics requires more time than is available to complete this thesis, and the TSE relies on other aircraft systems only for input of data. Further, the TSE does not directly control any aircraft function.

1.4. Equipment Used

Each of the Kalman filter implementations investigated in this thesis is evaluated by means of software written for this purpose, the Simulation for Optimal Filter Evaluation (SOFE) and associated plotting post-processor (SOFEPL) routines [6,7]. Each of these tools are available on the AFIT/ENG ELXSI computer. A brief discussion of the evaluation process follows; a more detailed description of SOFE and SOFEPL is presented in Chapters II to V.

SOFE is an EKF evaluation tool which was written to test the effects of varying the parameters or equations which comprise a given Kalman filter. Written in FORTRAN, SOFE consists of a set of fixed subroutines that exercise the filter and a set of user-defined subroutines which define the particular filter to be tested. A set of input parameters is varied for input/output control and filter functioning, depending on the data required and the filter parameters desired to be tested. SOFE then performs the number of Monte Carlo runs specified by the user and generates any output files requested for later evaluation [6].

SOFEPL is a plotting postprocessor for SOFE. When SOFE has generated time history data files for the number of specified Monte Carlo runs, SOFEPL generates plot files for the types of plots requested, based upon ensemble averages across all the runs [7].

1.5. Organization

A more in-depth coverage of AFTI/F-16 AMAS concepts and a brief presentation of the extended Kalman filtering techniques used in this thesis are provided in Chapter II. Chapter III develops a constant turn rate (CTR) EKF as a baseline model for the TSE. A Gauss-Markov (GM) acceleration EKF is developed in Chapter IV, as is the U-D factorization implementation of the GM filter. Results of Monte Carlo simulations and an analysis of the filters is accomplished in Chapter V. Chapter V also presents the conclusions of this study and recommendations.

II. Review of Current Knowledge

2.1 Introduction

This chapter presents a discussion of the portion of the AFTI/F-16 aircraft relative to this thesis, including sensors and the digital flight control system (DFCS). The concept of integrated fire/flight control (IFFC) systems is explored, and the EKF is introduced as the target state estimator (TSE) for the IFFC. The reader is expected to have a background in stochastic processes and Kalman filters (estimation theory); for a rigorous development, see Reference 4. This chapter concludes with a discussion of the algorithms used in this thesis to implement the Kalman filters.

2.2 AFTI/F-16 Aircraft

The AFTI/F-16 is a modified F-16 fighter aircraft which tests the integration of new technologies into fighter aircraft. These modifications allow the AFTI/F-16 to have much greater flexibility than conventional aircraft to perform its role as a testbed for new technology [2].

One obvious modification is the addition of two vertical canards to the front underside of the airframe (see Figure 1). These vertical canards provide aerodynamic control surfaces that allow the AFTI/F-16 to perform such maneuvers as direct side force and vertical lift without pitch maneuvering [10].

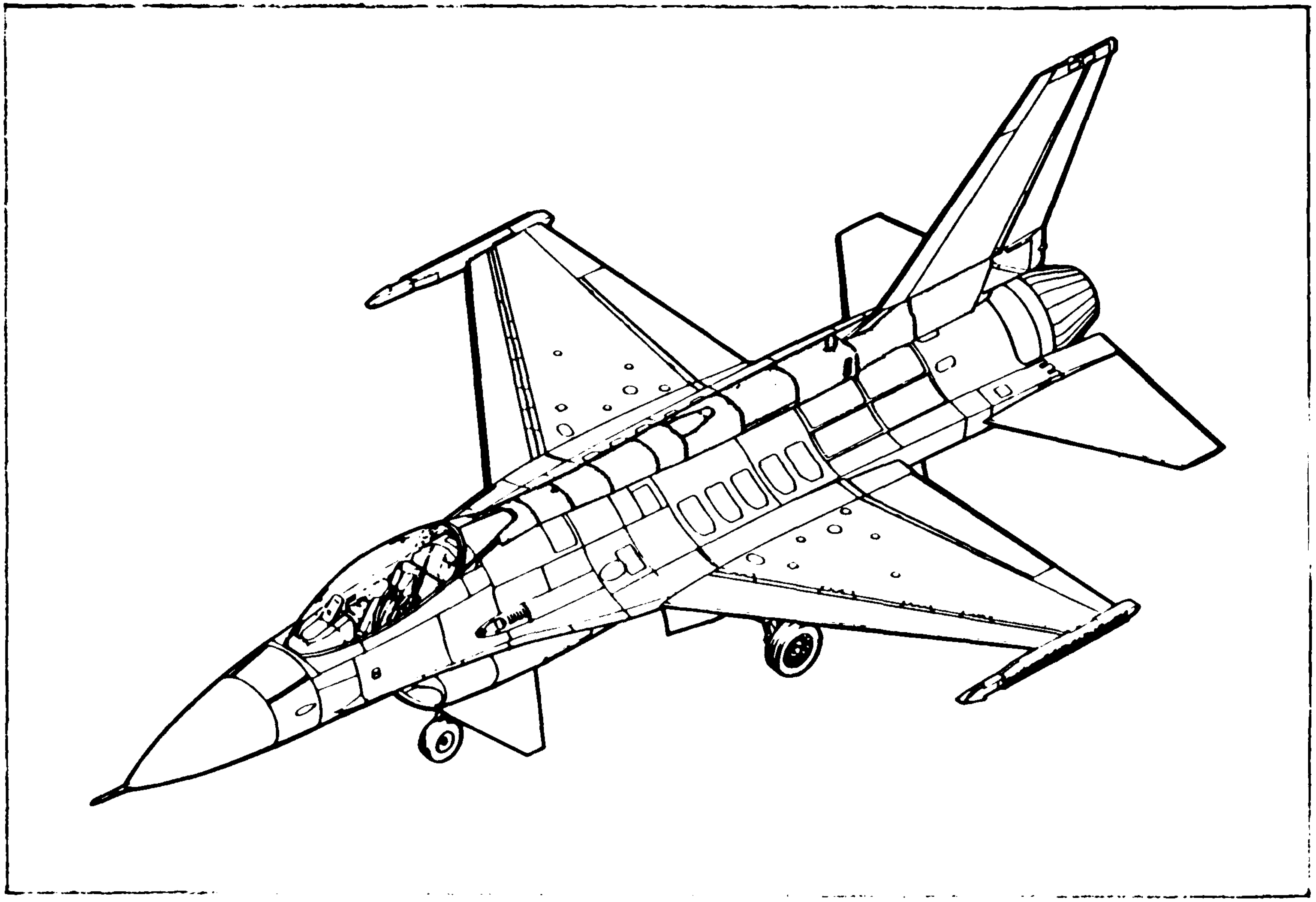


Figure 1. AFTI/F-16 Configuration [10]

Another feature of the aircraft is its total fly-by-wire control system, which uses triply-redundant digital computers to perform all flight control and fire control system functions. Changing system responses, aircraft configuration, or mission capabilities is accomplished by modifying the programs which run on the system computers [2].

2.2.1 Sensors

Sensors for tracking a target include the APG-66 radar system, the Sensor/Tracker Set (STS), and the pilot. For purposes of this thesis, the pilot is not included as a sensor.

The APG-66 radar system uses an integral Kalman filter to predict future target position and rates. This Kalman filter uses an inertial Cartesian coordinate frame to express the states of relative target position, relative target velocity, and total target acceleration; i.e., the radar states are

$$\hat{\underline{x}} = \begin{bmatrix} \underline{R} \\ \underline{V} \\ \underline{A}_T \end{bmatrix} \tag{2-1}$$

where

$\underline{R}$ is the (relative) target position vector,
$\underline{V}$ is the (relative) target velocity vector,
$\underline{A}$ is the target acceleration vector, and
T indicates total (inertial) quantities

as expressed in inertial (e.g., North/East/Down) coordinates.

Unfortunately, target state estimates from the radar are considered to be too noisy and inaccurate to be used for accurate gun pointing [10]. In fact, the least significant bit of the radar target position estimate represents 16 feet [10] (Note the performance of the STS later in this section).

The STS consists of sensors and a digital computer dedicated for its use to compute target position, velocity, and acceleration; i.e., as a target state estimator. The sensors are a FLIR and laser ranger which share common optics to eliminate sighting errors. The TSE currently employed in the STS is implemented in the same inertial Cartesian coordinate frame as the TSE for the radar and uses the same states as the radar (given in Equation (2-1)) [10].

Besides being used by other on-board computers (as described below), TSE target states are used to generate rate aiding commands to keep the FLIR/tracker head pointed at the target during target maneuvers. Target position states in the TSE have an accuracy specification of two feet (one-sigma). This accuracy ensures that the fire control computers can reliably predict future target position [10].

When a target is designated by the pilot, the radar system's target states and covariances are provided to the STS, which then begins tracking. Data from the INU are available directly to the TSE, and the TSE bases its updates on a "snapshot" of all sensor data. The update, then, carries a time tag so any computer on the aircraft can tell the precise time of the most recent update. This is important because the TSE updates at a 30 Hertz rate (limited by the FLIR scan rate) while all other computer systems on the aircraft operate at a 50 Hertz update rate [10].

2.2.2. Digital Flight Control System

The digital computers on the AFTI/F-16 perform all flight control and fire control functions. Figure II depicts the interaction between various components of the digital flight control system (DFCS). All sensor data are digitized and sent to one or more digital computers for processing. The flight control computers convert flight control commands to control surface commands so that the aircraft performs as desired [10].

2.3 Integrated Fire/Flight Control Concepts

When weapons were first carried on board aircraft, the pilot (or gunner) was the sensor, data processor, and controller for pointing and firing the weapon. Lead angles for pointing the aircraft or gun were computed by the pilot (or gunner) during the engagement, and were limited by his experience and reaction time. To aid the pilot in deciding when to fire and how to fly the aircraft, target and bullet prediction algorithms such as the lead-computing sight and the tracer algorithm were developed. The IFFC aids the pilot even more, by automating aircraft maneuvers and gun control commands. The aircraft positions itself (via the IFFC system) and fires the gun at the target designated by the pilot when an acceptable probability of kill exists [9].

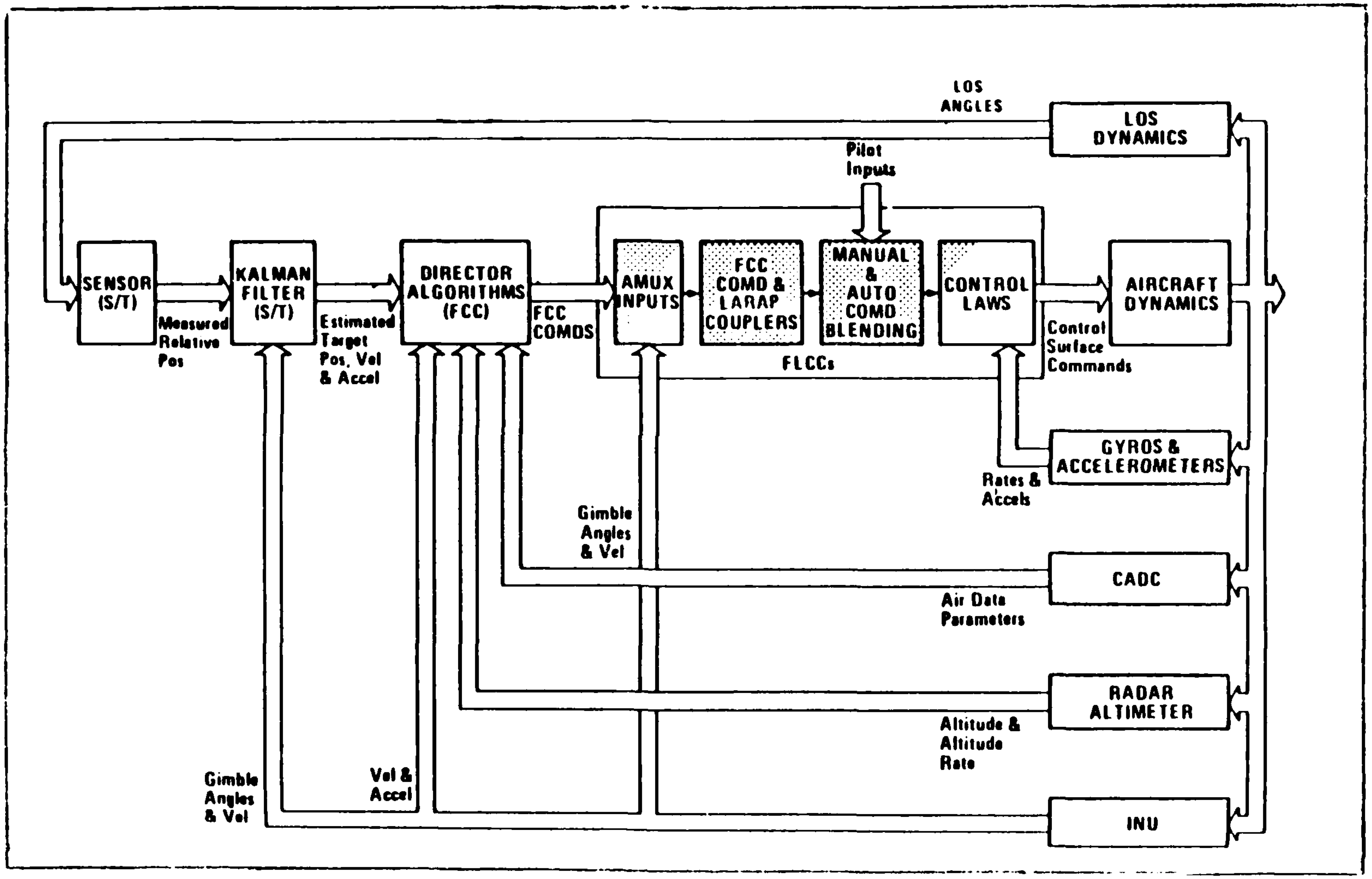


Figure 2. Digital Flight Control System [10]

In order to point the aircraft properly, the IFFC system computes bullet and target position one bullet time-of-flight (TOF) into the future. Corrections to the aircraft flight path are then computed to put the bullets on the target at the end of the bullet TOF. Future bullet positions can be found by using bullet trajectory algorithms or by integrating the bullet dynamics equations. Future target position can also be determined by integrating the target dynamics equations forward to the end of the bullet TOF [10].

In order to predict the future position of the target accurately, the IFFC system needs an accurate estimate of present target states. In the AFTI/F-16, this target state estimation is accomplished with an EKF in the STS [10]. This EKF is the subject of this thesis.

2.4 Kalman Filtering

This subsection presents the EKF equations applicable to this thesis, and illustrates why the conventional covariance updates are inadequate for small wordlength computers. An alternate covariance update algorithm, U-D factorization, is presented and used to implement a proposed EKF for this research. This subsection concludes with a discussion of the difference between the measurement frame and the state estimate frame used in the EKFs of this thesis.

2.4.1. Extended Kalman Filter Equations

The EKF equations for propagation and update [5:43,44] are presented below. For a full derivation of these equations, see references 4 and 5.

Assume that the system of interest is described by a stochastic process whose dynamics model is

$$\dot{\hat{\underline{x}}}(t) = \underline{f}[\hat{\underline{x}}(t),\underline{u}(t),t] + G(t)\underline{w}(t) \qquad (2-2)$$

where

$\underline{f}[\cdot,\cdot,\cdot]$ is a (possibly nonlinear) function of the state estimates, inputs, and time,

$\hat{\underline{x}}(\cdot)$ is the estimated state n-vector, and $\hat{\underline{x}}(t_o)$ is modeled as a Gaussian random n-vector with mean $\hat{\underline{x}}_o$ and covariance P_o,

$\underline{u}(\cdot)$ is an r-vector of known input functions,

$G(\cdot)$ is an n-by-s measurement noise input matrix, and

$\underline{w}(\cdot,\cdot)$ is a zero-mean white Gaussian s-vector process independent of $\hat{\underline{x}}(t_o)$ and of strength $Q(t)$, i.e., $E\{\underline{w}(t)\underline{w}^T(t+\tau)\} = Q(t)\delta(t)$

and the sampled-data (discrete-time) measurement is modeled as the m-vector process

$$\underline{z}(t_i) = \underline{h}[\hat{\underline{x}}(t_i),t_i] + \underline{v}(t_i) \qquad (2-3)$$

where

$\underline{h}(\cdot,\cdot)$ is a (possibly nonlinear) m-vector function of the states and time, and

$\underline{v}(\cdot,\cdot)$ is a zero-mean white Gaussian m-vector process, independent of $\hat{\underline{x}}(t_o)$ and $\underline{w}(\cdot,\cdot)$, and of covariance $R(t_i)$.

For propagation between measurements, the following two equations are integrated forward in time:

$$\dot{\hat{\underline{x}}}(t|t_i) = \underline{f}[\hat{\underline{x}}(t|t_i),\underline{u}(t),t] \tag{2-4}$$

$$\dot{P}(t|t_i) = F[t;\hat{\underline{x}}(t|t_i)]P(t|t_i) + P(t|t_i)F^T[t,\hat{\underline{x}}(t|t_i)] + G(t)Q(t)G^T(t) \tag{2-5}$$

where

F(·,·) is defined to be the n-by-n matrix of partial derivatives,

$$F \equiv \left.\frac{\partial\, \underline{f}[\underline{x},\underline{u}(t),t]}{\partial\, \underline{x}}\right|_{\underline{x}=\hat{\underline{x}}(t|t_i)} \tag{2-6}$$

For measurement update, the measurements $\underline{z}(t_i)$ are incorporated by the equations

$$K = P^-H^T[HP^-H^T + R]^{-1} \tag{2-7}$$

$$\hat{\underline{x}}^+ = \hat{\underline{x}}^- + K\{\underline{z}(t_i) - \underline{h}[\hat{\underline{x}}(t_i^-),t_i]\} \tag{2-8}$$

$$P^+ = P^- - KHP^- \tag{2-9}$$

where

H is defined to be the m-by-n matrix of partial derivatives,

$$H \equiv \left.\frac{\partial\, \underline{h}[\underline{x},t_i]}{\partial\, \underline{x}}\right|_{\underline{x}=\hat{\underline{x}}(t_i^-)} \tag{2-10}$$

$\underline{z}(t_i)$ is the measurement vector from sensors in actual applications or a truth model of the system for simulation evaluation,

and the time arguments have been omitted for clarity [$P^- = P(t_i^-)$, etc.].

Equation (2-7) calculates what is commonly called the Kalman gain of the filter and the term in braces ({·}) in Equation (2-8) is termed the residual. Equation (2-9) performs the conventional covariance update.

2.4.2. Wordlength Problems

Note that a matrix inversion is required to calculate the Kalman gain in Equation (2-7). For a matrix to be invertible, its determinant must not be zero (the matrix must not be singular). If computers had infinite wordlength, this would not be a problem, because the first term ($HP^{-}H^{T}$) is guaranteed to be at least positive semidefinite while the second term (R) is guaranteed (by definition) to be positive definite [4:216]. However, when a computer with a finite wordlength (as all digital computers are) is used, the filter may diverge or totally fail. Maybeck states:

> For instance, although it is theoretically impossible for the covariance matrix to have negative eigenvalues, such a condition can, and often does, result due to numerical computation using finite wordlength, especially when (1) the measurements are very accurate [eigenvalues of $R(t_i)$ are small relative to those of $P(t_i^-)$, this being accentuated by large eigenvalues in P_o] or (2) a linear combination of state vector components is known with great precision while other combinations are nearly unobservable... [4:368].

In general, updating the covariance matrix usually requires at least double precision arithmetic [4:368]. Even double precision arithmetic does not guarantee stability of the Kalman recursion equations; in fact, the conventional Kalman filter equations have been shown to be inherently numerically unstable [1:338].

Various alternate recursion relations have been developed which are inherently stable. Notable among these are various square root forms and U-D factorization algorithms [4]. The concept of the square root forms is to propagate and update the square root of the covariance matrix. Such a technique can yield twice the effective precision of the conventional equations in ill-conditioned problems, or the same accuracy with about half of the wordlength required for the conventional equations [4:369]. More importantly, the square root forms have been found to be numerically stable [1:338]. The U-D factorization form shares these desirable attributes, and has the added advantage that computationally time-consuming square roots are never explicitly evaluated [4:392].

2.4.3. U-D Factorization

The U-D factorization method expresses the covariance before and after update at time t_i as

$$P(t_i^-) = U(t_i^-)\ D(t_i^-)\ U^T(t_i^-) \qquad (2\text{-}11)$$

$$P(t_i^+) = U(t_i^+)\ D(t_i^+)\ U^T(t_i^+) \qquad (2\text{-}12)$$

where

U(·) is an upper diagonal, unitary matrix; i.e., all elements below the main diagonal are zero and all elements on the main diagonal are one, and

D(·) is a diagonal matrix; i.e., the only nonzero elements are on the diagonal.

The computations in Equations (2-11) and (2-12) are not actually accomplished; rather, algorithms are used to compute $\hat{x}$, U, and D directly instead of $\hat{x}$ and P.

Although the U and D matrices are not unique, a uniquely defined pair can be generated algorithmically. Several methods exist to generate the n-by-n U and D matrices; one such method [4:392] is presented below:

First, for the n-th column

$$D_{in} = \begin{cases} P_{nn} & i = n \\ 0 & i \neq n \end{cases}$$

$$U_{in} = \begin{cases} 1 & i = n \\ P_{in}/D_{nn} & i = n-1,\ n-2,\ \ldots,\ 1 \end{cases} \qquad (2\text{-}13)$$

Then, for the remaining columns, for j=n-1, n-2, ..., 1

$$D_{ij} = \begin{cases} P_{jj} - \sum_{k=j+1}^{n} D_{kk} U_{jk}^{2} & i = n \\ 0 & i \neq n \end{cases} \qquad (2\text{-}14)$$

$$U_{ij} = \begin{cases} 0 & i > j \\ 1 & i = j \\ \left(P_{ij} - \sum_{k=j+1}^{n} D_{kk} U_{ik} U_{jk}\right)/D_{jj} & i = j-1,\ \ldots,\ 1 \end{cases}$$

Equations (2-13) and (2-14) are normally used only for initialization of U_o and D_o; propagation and update algorithms (such as those below) compute U and D factors.

Measurement update using the U-D factorization filter is first presented for the scalar case [4:394], then generalized to the vector case below.

For a scalar update, compute

$$\begin{aligned} \underline{f} &= U^T(t_i^-)\underline{H}^T(t_i) \\ v_j &= D_{jj}(t_i^-)f_j \qquad j=1,\ 2,\ \ldots,\ n \qquad (2\text{-}15) \\ a_o &= R \end{aligned}$$

Then, for k=1, 2, ..., n, calculate

$$\begin{aligned} a_k &= a_{k-1} + f_k v_k \\ D_{kk}(t_i^+) &= D_{kk}(t_i^-)a_{k-1}/a_k \\ b_k &\leftarrow v_k \\ p_k &= -f_k/a_{k-1} \\ U_{jk}(t_i^+) &= U_{jk}(t_i^-) + b_j p_k \\ b_j &\leftarrow b_j + U_{jk}(t_i^-)v_k \qquad j=1,\ 2,\ \ldots,\ k-1 \end{aligned} \tag{2-16}$$

where $\leftarrow$ denotes replacement, or 'writing over' old variables for efficiency.

Vector measurement updates are processed component by component, assuming the R matrix is diagonal (if the R matrix is not diagonal, a transformation of variables is required so that it is made diagonal; the diagonal requirement is discussed further below) [4:394]. When $U(t^+)$ and $D(t^+)$ have been calculated, the Kalman gain can be calculated from

$$\underline{K}(t_i) = \underline{b}/a_n \tag{2-17}$$

and the states updated using Equation (2-8) [4:394].

Time propagation of the U-D factors involves many concepts that are beyond the scope of this thesis. One method for time propagation of U and D is [4:396-397]:

Form the matrices $Y(t_{i+1}^-)$ and $\underline{\tilde{D}}(t_{i+1}^-)$ by augmenting:

$$Y(t_{i+1}^-) = [\Phi(t_{i+1},t_i)U(t_i^+) \mid G_d(t_i)] \tag{2-18}$$

$$\underline{\tilde{D}}(t_{i+1}^-) = \left[\begin{array}{c|c} D(t_i^+) & 0 \\ \hline 0 & Q_d(t_i) \end{array}\right] \tag{2-19}$$

where

$\Phi(t_{i+1},t_i)$ is the state transition matrix which propagates the states forward in time from t_i to t_{i+1},

G_d is the n-by-s discrete time noise input matrix, and

Q_d is the s-by-s noise matrix which has statistics identical to Q at the discrete sample times [4:171]:

$$Q_d(t_i) = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1},\tau)G(\tau)Q(\tau)G^T(\tau)\Phi^T(t_{i+1},\tau)d\tau \qquad (2\text{-}20)$$

Then, initialize n vectors, each of dimension (n + s), through

$$[\ \underline{a}_1 \mid \underline{a}_2 \mid \dots \mid \underline{a}_n\ ] = Y^T(t_{i+1}^-) \qquad (2\text{-}21)$$

and iterate on the following relations for k=n, n-1, ..., 1:

$$\begin{aligned}
\underline{c}_k &= \underline{\tilde{D}}(t_{i+1}^-)\underline{a}_k \\
&\quad \{c_{kj} = \tilde{D}_{jj}(t_{i+1}^-)a_{kj},\ j=1,\ 2,\ \dots,\ n+s\} \\
D_{kk}(t_{i+1}^-) &= \underline{a}_k^T\underline{c}_k \\
\underline{d}_k &= \underline{c}_k/D_{kk}(t_{i+1}^-) \qquad (2\text{-}22) \\
U_{jk}(t_{i+1}^-) &= \underline{a}_j^T\underline{d}_k \qquad j=1,\ 2,\ \dots,\ k-1 \\
\underline{a}_j &\leftarrow \underline{a}_j - U_{jk}(t_{i+1}^-)\underline{a}_k
\end{aligned}$$

Again, ← denotes 'writing over' old variables. The state estimate is given by

$$\hat{\underline{x}}(t_{i+1}^-) = \Phi(t_{i+1},t_i)\hat{\underline{x}}(t_i^+) + B_d(t_i)\underline{u}(t_i) \qquad (2\text{-}23)$$

where

B_d is the discrete-time input matrix, defined by [4:171]

$$B_d = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1},\tau)B(\tau)\,d\tau \qquad (2\text{-}24)$$

2.4.4. Reference Frames: Measurement vs. State

Whenever two quantities are summed they must be expressed in the same reference frame for the sum to be valid. However, in Equation (2-7), the quantities utilized by the STS are expressed in two different coordinate frames. The TSE states and covariances are expressed in the inertial reference frame, while the measurements and their variances are known only in the LOS frame. If the frames are closely aligned at a given time (i.e., the sensor head is pointed along the "north"-axis of the inertial frame), the variances along each axis are nearly the same in either frame [8:40].

However, since the reference frames are not generally aligned, the covariances expressed in the LOS frame is skewed in the inertial frame (see Figure 2.3 for a simplified example). Therefore, a change of reference frame must be performed for each update cycle [9]. This change of frame may take one of two forms:

(1) The covariances and state estimates are rotated from the inertial frame into the measurement frame after propagation to t_i^- but before measurement update at t_i. This rotation, as used in this thesis, is based upon an Euler

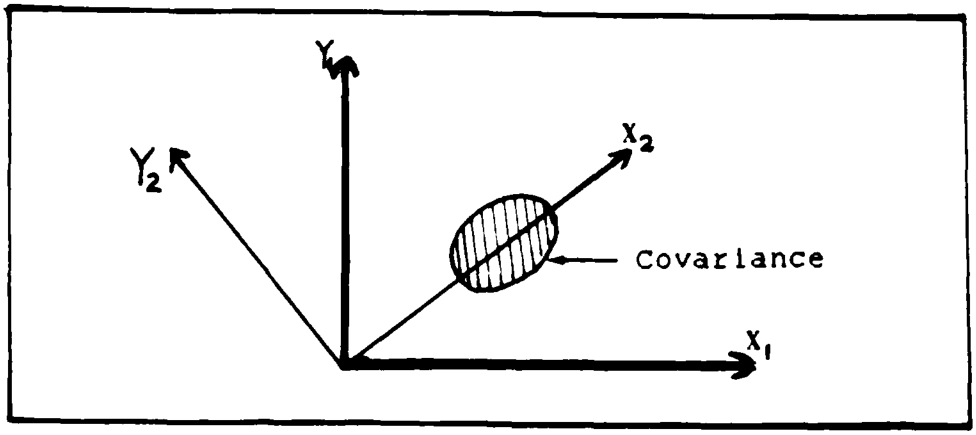


Figure 3. Covariances in Two Reference Frames

angle transformation developed in Appendix A [8]. The covariances and state estimates must then be rotated back into the inertial frame after measurement update for normal propagation, as in Equation (2-5).

(2) The measurement noise matrix R is rotated from the LOS frame into the inertial frame before measurement update occurs.

The first option above involves the use of two matrix transformations. Because the transformations are inverse (and transpose) operations, calculations can be greatly simplified, as shown in Chapter III.

The second option involves only one transformation, at first glance. However, after a diagonal R matrix is rotated into the inertial frame it would not, in general, be diagonal. Since vector updates in the U-D factorization algorithm require the R matrix to be diagonal, a change of variables would have to be done to accommodate the update algorithm. These additional rotations are exactly the rotations required in the first option above; therefore, the method shown in the first option is used in this thesis to rotate the covariances and states during update.

2.4.5. Filter Implementations

Several models have been used in the literature to represent target motion in an air-to-air attack scenario [8]. Several models use various techniques to calculate the acceleration state as a function of time and integrate successively to obtain the velocity and position. The states themselves may be expressed in a variety of reference frames.

One such model is the Gauss-Markov acceleration inertial coordinate (GM) filter. In the GM model, the accelerations are calculated using a first-order Gauss-Markov acceleration model, but do not account for persistent (non-zero mean) accelerations. Although the GM model does not reflect the real-world system with great accuracy [8:17], the GM filter has the advantage of linear dynamics.

Another candidate model is the constant inertial target turn rate constant-speed inertial coordinate (CTR) filter. In the CTR filter, the accelerations are calculated based upon a constant speed target going through a constant turn rate, planar turn. The acceleration model of the CTR filter approximates actual target accelerations more closely than the GM filter [8:3,22], but the dynamics model is nonlinear, as is shown in Chapter III.

Because the CTR filter more accurately models an aerial engagement, it is used as a baseline model for this thesis. However, due to its complexity, the CTR filter is impractical for use in the TSE. The equations involved in propagating and updating the states and covariances require much more time and memory in the CTR filter than in the GM filter, as is evident from the dynamics models of the filters (see Chapters III and IV). For this reason, a modified form of the GM filter is developed in Chapter IV of this thesis as the proposed model for the TSE.

2.5. Summary

This chapter presented a brief overview of AFTI/F-16 systems and IFFC concepts to motivate target state estimator implementation on this aircraft. Also, the Kalman filtering techniques presented are used as tools for implementing target state estimators in the next two chapters.

III. Constant Turn Rate Filter

3.1. Introduction

The constant turn rate (CTR) EKF is presented in four parts in this chapter. These four parts are the dynamics model, the measurement model, the noise models, and specific implementation on SOFE.

The CTR filter is a complex filter (as is shown below) which closely models the real-world system [8]. Because the CTR filter so closely models the real-world system, it is used as a baseline filter for comparison purposes and few simplifying assumptions are made to reduce its complexity.

3.2. Filter Dynamics Model

Implementation of the CTR filter dynamics model involves the evaluation of the $\underline{f}$ and $\underline{w}$ vectors and the G matrix of Equation (2-2). The $\underline{x}$ vector (the estimated target state) is defined by Equation (2-1), expanded for clarity:

$$\hat{\underline{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_7 \\ \hat{x}_8 \\ \hat{x}_9 \end{bmatrix} \qquad (3\text{-}1)$$

where

$\hat{x}_1$, $\hat{x}_2$, and $\hat{x}_3$ are the relative target positions along the 1, 2, and 3 axes, respectively, of the inertial coordinate frame (e.g., North, East, and Down),

$\hat{x}_4$, $\hat{x}_5$, and $\hat{x}_6$ are the relative target velocities along the 1, 2, and 3 axes, respectively, of the inertial coordinate frame, and

$\hat{x}_7$, $\hat{x}_8$, and $\hat{x}_9$ are the total target accelerations along the 1, 2, and 3 axes, respectively, of the inertial coordinate frame.

The differential equations describing the position states as a function of the velocity states are written as

$$\dot{\hat{x}}_1 = \hat{x}_4 \tag{3-2}$$

$$\dot{\hat{x}}_2 = \hat{x}_5 \tag{3-3}$$

$$\dot{\hat{x}}_3 = \hat{x}_6 \tag{3-4}$$

The differential equations describing the velocity states as a function of accelerations are written as

$$\dot{\hat{x}}_4 = \hat{x}_7 - a_1 \tag{3-5}$$

$$\dot{\hat{x}}_5 = \hat{x}_8 - a_2 \tag{3-6}$$

$$\dot{\hat{x}}_6 = \hat{x}_9 - a_3 \tag{3-7}$$

where a_1, a_2, and a_3 are the accelerations of the attacker along the 1, 2, and 3 axes, respectively, of the inertial coordinate frame.

The differential equations describing the acceleration states are written as

$$\dot{\hat{x}}_7 = -\|\underline{\omega}\|^2(\hat{x}_4 + v_1) + w_1 \tag{3-8}$$

$$\dot{\hat{x}}_8 = -\|\underline{\omega}\|^2(\hat{x}_5 + v_2) + w_2 \tag{3-9}$$

$$\dot{\hat{x}}_9 = -\|\underline{\omega}\|^2(\hat{x}_6 + v_3) + w_3 \tag{3-10}$$

where

v_1, v_2, and v_3 are the velocities of the attacker, provided by the INU, along the 1, 2, and 3 axes, respectively, of the inertial coordinate system (thus, $\hat{x}_i+v_j$ is the total target velocity in the j-direction),

w_1, w_2, and w_3 are zero-mean, white Gaussian noises, independent of each other and of $\hat{\mathbf{x}}$, and

the term $\|\boldsymbol{\omega}\|^2$ is the square of the magnitude of the angular velocity, calculated as

$$\|\boldsymbol{\omega}\|^2 = \frac{A_1^2 + A_2^2 + A_3^2}{A_4^2} \tag{3-11}$$

where

$$A_1 = (\hat{x}_5 + v_2)\hat{x}_9 - (\hat{x}_6 + v_3)\hat{x}_8 \tag{3-12}$$

$$A_1 = (\hat{x}_6 + v_3)\hat{x}_7 - (\hat{x}_4 + v_1)\hat{x}_9 \tag{3-13}$$

$$A_3 = (\hat{x}_4 + v_1)\hat{x}_8 - (\hat{x}_5 + v_2)\hat{x}_7 \tag{3-14}$$

$$A_4 = (\hat{x}_4+v_1)^2 + (\hat{x}_5+v_2)^2 + (\hat{x}_6+v_3)^2 \tag{3-15}$$

The dynamics equations for each of the states are expressed in matrix form as

$$\dot{\hat{\mathbf{x}}} = \begin{bmatrix} \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_7 - a_1 \\ \hat{x}_8 - a_2 \\ \hat{x}_9 - a_3 \\ -\|\boldsymbol{\omega}\|^2(\hat{x}_4 + v_1) \\ -\|\boldsymbol{\omega}\|^2(\hat{x}_5 + v_2) \\ -\|\boldsymbol{\omega}\|^2(\hat{x}_6 + v_3) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} \tag{3-16}$$

which is in the form of Equation (2-2). Using Equation (3-16), the partial-derivative matrix F is (see Equation (2-6))

$$F = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \\ 0 & 0 & 0 & F_7 & F_8 & F_9 & F_{10} & F_{11} & F_{12} \\ 0 & 0 & 0 & F_{13} & F_{14} & F_{15} & F_{16} & F_{17} & F_{18} \end{bmatrix} \qquad (3\text{-}17)$$

where

$$F_1 = -\left(\|\underline{\omega}\|^2 + \frac{2(\hat{x}_4+v_1)[A_4(A_3\hat{x}_8-A_2\hat{x}_9)-2(A_1^2+A_2^2+A_3^2)(\hat{x}_4+v_1)]}{A_4^3} \right) \qquad (3\text{-}18)$$

$$F_2 = -\frac{2(\hat{x}_4+v_1)[A_4(A_1\hat{x}_9-A_3\hat{x}_7) - 2(A_1^2+A_2^2+A_3^2)(\hat{x}_5+v_2)]}{A_4^3} \qquad (3\text{-}19)$$

$$F_3 = -\frac{2(\hat{x}_4+v_1)[A_4(A_2\hat{x}_7-A_1\hat{x}_8) - 2(A_1^2+A_2^2+A_3^2)(\hat{x}_6+v_3)]}{A_4^3} \qquad (3\text{-}20)$$

$$F_4 = -\frac{2(\hat{x}_4+v_1)[A_2(\hat{x}_6+v_3) - A_3(\hat{x}_5+v_2)]}{A_4^2} \qquad (3\text{-}21)$$

$$F_5 = - \frac{2(\hat{x}_4+v_1)[\ A_3(\hat{x}_4+v_1) - A_1(\hat{x}_6+v_3)\ ]}{A_4^2} \tag{3-22}$$

$$F_6 = - \frac{2(\hat{x}_4+v_1)[\ A_1(\hat{x}_5+v_2) - A_2(\hat{x}_4+v_1)\ ]}{A_4^2} \tag{3-23}$$

$$F_7 = - \frac{2(\hat{x}_5+v_2)[A_4(A_3\hat{x}_8-A_2\hat{x}_9)-2(A_1^2+A_2^2+A_3^2)(\hat{x}_4+v_1)]}{A_4^3} \tag{3-24}$$

$$F_8 = - \left(\|\omega\|^2 + \frac{2(\hat{x}_5+v_2)[A_4(A_1\hat{x}_9-A_3\hat{x}_7)-2(A_1^2+A_2^2+A_3^2)(\hat{x}_5+v_2)]}{A_4^3} \right) \tag{3-25}$$

$$F_9 = - \frac{2(\hat{x}_5+v_2)[\ A_4(A_2\hat{x}_7-A_1\hat{x}_8) - 2(A_1^2+A_2^2+A_3^2)(\hat{x}_6+v_3)\ ]}{A_4^3} \tag{3-26}$$

$$F_{10} = - \frac{2(\hat{x}_5+v_2)[\ A_2(\hat{x}_6+v_3) - A_3(\hat{x}_5+v_2)\ ]}{A_4^2} \tag{3-27}$$

$$F_{11} = - \frac{2(\hat{x}_5+v_2)[\ A_3(\hat{x}_4+v_1) - A_1(\hat{x}_6+v_3)\ ]}{A_4^2} \tag{3-28}$$

$$F_{12} = - \frac{2(\hat{x}_5+v_2)[\ A_1(\hat{x}_5+v_2) - A_2(\hat{x}_4+v_1)\ ]}{A_4^2} \tag{3-29}$$

$$F_{13} = - \frac{2(\hat{x}_6+v_3)[\ A_4(A_3\hat{x}_8-A_2\hat{x}_9) - 2(A_1^2+A_2^2+A_3^2)(\hat{x}_4+v_1)\ ]}{A_4^3} \tag{3-30}$$

$$F_{14} = - \frac{2(\hat{x}_6+v_3)[\ A_4(A_1\hat{x}_9-A_3\hat{x}_7) - 2(A_1^2+A_2^2+A_3^2)(\hat{x}_5+v_2)\ ]}{A_4^3} \tag{3-31}$$

$$F_{15} = -\left(\|\underline{\omega}\|^2 + \frac{2(\hat{x}_6+v_3)[A_4(A_2\hat{x}_7 - A_1\hat{x}_8) - 2(A_1^2 + A_2^2 + A_3^2)(\hat{x}_6+v_3)]}{A_4^3} \right) \qquad (3\text{-}32)$$

$$F_{16} = -\frac{2(\hat{x}_6-v_3)[\ A_2(\hat{x}_6+v_3) - A_3(\hat{x}_5+v_2)\ ]}{A_4^2} \qquad (3\text{-}33)$$

$$F_{17} = -\frac{2(\hat{x}_6+v_3)[\ A_3(\hat{x}_4+v_1) - A_1(\hat{x}_6+v_3)\ ]}{A_4^2} \qquad (3\text{-}34)$$

$$F_{18} = -\frac{2(\hat{x}_6+v_3)[\ A_1(\hat{x}_5+v_2) - A_2(\hat{x}_4+v_1)\ ]}{A_4^2} \qquad (3\text{-}35)$$

Note that the $\underline{u}$ vector of Equation (2-2), the velocity and acceleration of the attacker, is imbedded in Equations (3-16) and (3-17). Also, the product of G and $\underline{w}$ is given as the last (column) vector of Equation (3-16).

3.3. Measurement Equations

Four measurements are available from the STS. Each measurement is discussed below as it relates to the states, then the H matrix of Equation 2-10 is presented.

3.3.1. Range Measurement

A laser range finder in the STS provides the range measurement. The range, expressed in the LOS coordinate frame, is computed as the time required for a pulsed beam of light to travel to the target and back, divided by twice the speed of light. The laser ranger in the STS has a range accuracy of 2.2 meters (one-sigma) [10]. From spherical geometry, the range to the target is expressed as the square root of the sum of the squares of the relative position states in either the inertial or the LOS coordinate frame; i.e., using state variables,

$$R = (\hat{x}_1^2 + \hat{x}_2^2 + \hat{x}_3^2)^{(1/2)} \tag{3-36}$$

3.3.2. Velocity Measurement

The relative target velocity along the LOS is measured by the APG-66 radar system by measuring the Doppler shift of the pulse of electromagnetic energy reflected from the target. The accuracy of the velocity measurement furnished by the radar system is one foot per second (one-sigma) [10]. The velocity measurement as a function of the states is computed, in either the inertial or LOS coordinate frame, by forming the dot product of the relative position and relative velocity vectors:

$$V = (\underline{P} \bullet \underline{V}) / R \tag{3-37}$$

where

V is the (signed) scalar relative velocity,

$\underline{P}$ is the relative position vector,

$$\underline{P} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} \tag{3-38}$$

$\underline{V}$ is the relative velocity vector,

$$\underline{V} = \begin{bmatrix} \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \end{bmatrix} \tag{3-39}$$

and R is the range to the target, defined by Equation (3-36).

Equation (3-37) is written using state variables as:

$$V = (\hat{x}_4\hat{x}_1 + \hat{x}_5\hat{x}_2 + \hat{x}_6\hat{x}_3) / R \qquad (3\text{-}40)$$

3.3.3. Angle Measurements

The FLIR in the STS provides a measurement of angular errors with respect to the boresight of the FLIR head. The boresight, or sensor head position, angles with respect to the aircraft body are known from position sensors in the STS, and the INU provides aircraft attitude data directly to the STS. Therefore, all angles between the inertial frame and the LOS frame are known when a sample occurs, and the azimuth and elevation angles from the attacker to the target are calculated as a 'snapshot' of these angles. The STS measures the azimuth and elevation angles with a precision of 1.13 milliradians (one-sigma) [10]. This figure includes all error sources: INU measurement errors, sensor-head alignment errors, FLIR pixel errors, etc. [10]. The angle measurements are generated from the state estimates by rotating the state estimates into the sensor reference frame and comparing angular distances, as explained below.

Figure IV describes the angle relationships present during a measurement. The Euler angle relationships are developed in Appendix A; the matrix which rotates quantities from the inertial frame to the LOS frame is

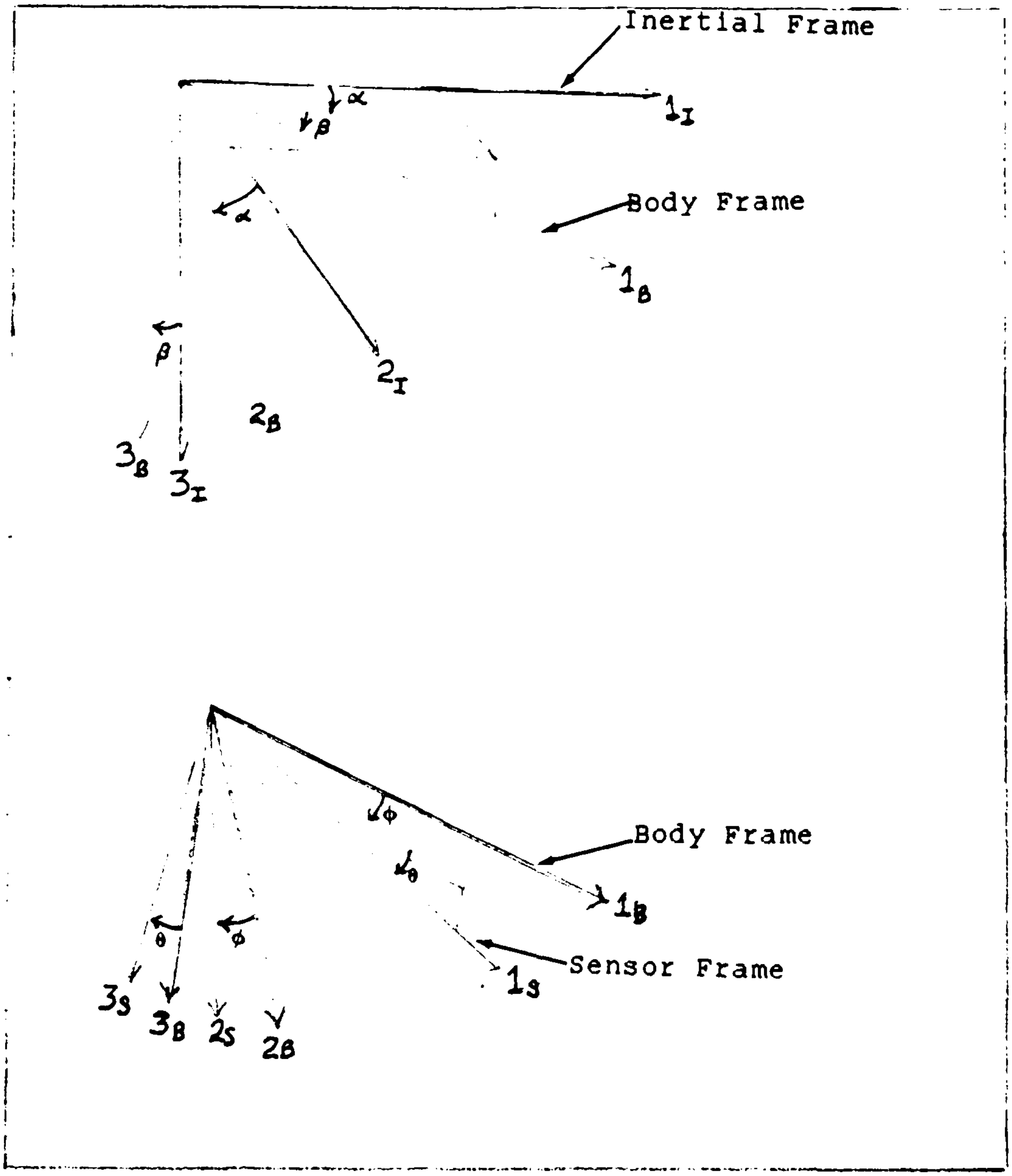


Figure 4. Measurement Angles

$$T_L^I = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \qquad (3\text{-}41)$$

where

$$T_{11} = \cos\theta \cos\phi \cos\alpha \cos\beta - \sin\theta \cos\phi \sin\alpha - \sin\phi \cos\alpha \sin\beta \qquad (3\text{-}42)$$

$$T_{12} = \cos\theta \cos\phi \sin\alpha \cos\beta + \sin\theta \cos\phi \cos\alpha - \sin\phi \sin\alpha \sin\beta \qquad (3\text{-}43)$$

$$T_{13} = -\cos\theta \cos\phi \sin\beta - \sin\phi \cos\beta \qquad (3\text{-}44)$$

$$T_{21} = -\sin\theta \cos\alpha \cos\beta - \cos\theta \sin\alpha \qquad (3\text{-}45)$$

$$T_{22} = -\sin\theta \sin\alpha \cos\beta + \cos\theta \cos\alpha \qquad (3\text{-}46)$$

$$T_{23} = \sin\theta \sin\alpha \qquad (3\text{-}47)$$

$$T_{31} = \cos\theta \sin\phi \cos\alpha \cos\beta - \sin\theta \sin\phi \sin\alpha + \cos\phi \cos\alpha \sin\beta \qquad (3\text{-}48)$$

$$T_{32} = \cos\theta \sin\phi \cos\beta + \sin\theta \sin\phi \cos\alpha + \cos\phi \sin\alpha \sin\beta \qquad (3\text{-}49)$$

$$T_{33} = -\cos\theta \sin\phi \sin\beta + \cos\phi \cos\beta \qquad (3\text{-}50)$$

where

θ is the first Euler angle from the inertial axes to the body axes (the 'azimuth' of body attitude),

ϕ is the second Euler angle from the inertial axes to the body axes (the 'elevation' of body attitude),

α is the first Euler angle from the body axes to the sensor-head axes (the azimuth angle of the sensor head relative to the body), and

β is the second Euler angle from the body axes to the sensor-head axes (the elevation angle of the sensor head relative to the body)

as defined in Appendix A.

The position state estimates are rotated into the sensor frame by

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix}^L = T_L^I \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} \qquad (3\text{-}51)$$

and the tangents of the azimuth and elevation angles calculated as

$$\tan(a) = \hat{x}_2^L \; / \; \hat{x}_1^L \qquad (3\text{-}52)$$

$$\tan(e) = \hat{x}_3^L \; / \; \hat{x}_1^L \qquad (3\text{-}53)$$

Note that there is no need to correct the tangent function for a negative term in the denominators of the last two equations; the denominators in either equation cannot be negative if the sensor is tracking the target within 90 degrees. The tracker should, in fact, be pointing (tracking) the target within a few tenths of degrees, if operating properly.

3.3.4. The Measurement Matrix

To form the measurement vector, $\underline{h}(\hat{\underline{x}},\underline{u},t)$, of Equation (2-3), Equations (3-36), (3-40), (3-52) and (3-53) are combined:

$$h = \begin{bmatrix} \text{Eq. (3-36)} \\ \text{Eq. (3-40)} \\ \text{Eq. (3-52)} \\ \text{Eq. (3-53)} \end{bmatrix} \tag{3-54}$$

Evaluation of the partial-derivative matrix, H, of Equation (2-10), is straightforward for the range and velocity measurements. However, the transformation in the angle measurements is a function of the position states, and a fully developed H matrix requires partial differentiation of terms of the rotation matrix T_L^I with respect to the state vector. The terms involving differentiation of the rotation matrix are assumed to be small compared to the remaining terms, thus avoiding this differentiation and the resulting filter complexity. This assumption has been shown [8:52] to be valid. The H matrix is thus computed as

$$H = \begin{bmatrix} H_1 & H_2 & H_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ H_4 & H_5 & H_6 & H_7 & H_8 & H_9 & 0 & 0 & 0 \\ H_{10} & H_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ H_{12} & 0 & H_{13} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3-55}$$

where

$$H_1 = \hat{x}_1^L / R \tag{3-56}$$

$$H_2 = \hat{x}_2^L / R \tag{3-57}$$

$$H_3 = \hat{x}_3^L / R \tag{3-58}$$

$$H_4 = [R^2 \hat{x}_4^L - (\hat{x}_4^L \hat{x}_1^L + \hat{x}_5^L \hat{x}_2^L + \hat{x}_6^L \hat{x}_3^L) \hat{x}_1^L] / R^3 \tag{3-59}$$

$$H_5 = [R^2 \hat{x}_5^L - (\hat{x}_4^L\ \hat{x}_1^L + \hat{x}_5^L\ \hat{x}_2^L + \hat{x}_6^L\ \hat{x}_3^L\ )\hat{x}_2^L\ ] \ /\ R^3 \qquad (3\text{-}60)$$

$$H_6 = [R^2 \hat{x}_6^L - (\hat{x}_4^L\ \hat{x}_1^L + \hat{x}_5^L\ \hat{x}_2^L + \hat{x}_6^L\ \hat{x}_3^L\ )\hat{x}_3^L\ ] \ /\ R^3 \qquad (3\text{-}61)$$

$$H_7 = \hat{x}_1^L \ /\ R \qquad (3\text{-}62)$$

$$H_8 = \hat{x}_2^L \ /\ R \qquad (3\text{-}63)$$

$$H_9 = \hat{x}_3^L \ /\ R \qquad (3\text{-}64)$$

$$H_{10} = -\hat{x}_2^L \ /\ (\hat{x}_1^L\ )^2 \qquad (3\text{-}65)$$

$$H_{11} = 1 \ /\ \hat{x}_1^L \qquad (3\text{-}66)$$

$$H_{12} = -\hat{x}_3^L \ /\ (\hat{x}_1^L\ )^2 \qquad (3\text{-}67)$$

$$H_{13} = 1 \ /\ \hat{x}_1^L \qquad (3\text{-}68)$$

with

$$R = [(\hat{x}_1^L\ )^2 + (\hat{x}_2^L\ )^2 + (\hat{x}_3^L\ )^2]^{(1/2)} \qquad (3\text{-}69)$$

3.4. Noise Models

The noise models used in the CTR filter are divided into two categories and discussed separately. The two categories are dynamics driving noises, which comprise the $\underline{w}$ vector (with strength matrix Q) of Equation (2-5), and measurement noises, which comprise the $\underline{v}$ vector (with covariance matrix R) of Equation (2-7).

3.4.1. Dynamics Noises

The CTR filter acceleration states are driven by noises w_1, w_2, and w_3 which are assumed to be zero-mean, white, Gaussian, and uncorrelated with each other as well as the states. The strengths q_1, q_2, and q_3, respectively, of these noises are assumed to be equal; i.e., the target is assumed to be equally capable of acceleration in any direction. All accelerations are assumed to be less than 9 g's, which is treated as a two-sigma value to account for 95 percent of all expected accelerations. Thus, the strength of the noises (σ^2) is approximately 21,000 feet^2/second^5. If the G matrix of Equation (2-2) is assumed to be a 9x9 identity matrix, the Q matrix is

$$Q = \left[\begin{array}{c|ccc} \multicolumn{4}{c}{0_{6\times9}} \\ \hline & q_1 & 0 & 0 \\ 0_{3\times6} & 0 & q_2 & 0 \\ & 0 & 0 & q_3 \end{array}\right] \tag{3-70}$$

where $q_i = 21000$, $i = 1,2,3$.

3.4.2. Measurement Noises

Since real sensors are imperfect, the measurements available from them are corrupted by noise. These discrete-time noises are assumed to be zero mean, white, Gaussian, and uncorrelated with each other as well as the states. The zero mean, white, and Gaussian assumptions do an adequate job of modeling modern sensors, and the noises are uncorrelated if the measurements are made by independent devices. Although the devices performing the measurements are not totally independent, especially for the angle measurements, any dependence is assumed to have negligible effects on the measurements taken. If the measurements are not independent, a change of variables is done (see Section 2.4.3) to effect independence. These assumptions yield a diagonal R matrix,

$$R = \begin{bmatrix} R_1 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 \\ 0 & 0 & R_3 & 0 \\ 0 & 0 & 0 & R_4 \end{bmatrix} \quad (3\text{-}71)$$

where

R_1, R_2, R_3, and R_4 are the variances of the measurement noises for the range, range rate, and tangents of the azimuth and elevation angles, respectively.

Because the standard deviations for the angle measurements are small, the standard deviations for the tangents of the angles are approximately equal to the angular standard deviations by the small angle approximation. The measurement accuracies presented in Section 3.3 are one-sigma values; the accuracies and variances are listed in Table I.

Table I

Measurement Accuracies and Variances

Measurement	Accuracy (1σ)	Variance (σ^2)
Range	6.7 feet	45 feet^2
Range rate	1 foot/sec	1 $\text{feet}^2/\text{sec}^2$
Tangent (az)	1.13 mrad	$1.2769\text{X}10^{-6}\text{rad}^2$
Tangent (el)	1.13 mrad	$1.2769\text{X}10^{-6}\text{rad}^2$

3.5. CTR Filter Implementation

Implementing the CTR filter requires a slight modification to the general SOFE routines as well as development of filter-specific subroutines. A modification to the general SOFE routines is required so that the rotations described in Section 2.4.4 can be performed before and after each update. The filter-specific subroutines define the filter dynamics model, the measurement model, and the noise models, and provide input of filter variables to the program [6]. SOFE contains many features to enhance user interaction and output, which are explained in detail in Reference 6.

SOFE exercises the filter model by integrating the dynamics equations forward in time to the update time via a fifth order Kutta-Merson technique, then performing the update via a Carlson covariance square root algorithm. The integration/update cycle is repeated until the final time specified by the user is reached. The entire process from initial time to final time, called a run, is then repeated as many times as the user desires. Successive runs use a different random number sequence for the simulated input noises, so averaging the outputs from a sufficiently large number of runs can provide accurate performance sample statistics of a certain filter configuration [6]. This ensemble averaging is done by SOFEPL [7]. Results of the CTR filter runs are presented in Chapter V. Before the GM filter is presented, the reference frame rotation is discussed.

In order to perform the rotations described in Section 2.4. , the sensor head position with respect to the inertial frame must be known. In the actual system, the sensor head position is given by two Euler angle rotations: The first, from the inertial frame to the aircraft body axes, is determined by the INU; the second, from the aircraft body axes to the sensor head axes, is determined by angle resolvers in the sensor head mount [10]. In this thesis the attacker is assumed to be benign (as discussed in Section 1.3), so the first Euler angle rotation is eliminated. The second set of Euler angles is initialized so that the sensor head is pointed directly at the (estimated) target position, then incrementally rotated at each sample time by the following algorithm, which is used in the current implementation of the TSE [10]:

1) Rotate the inertial target position (from the truth model, or external trajectory) into the current LOS frame:

$$\underline{p}_L = T_L^I \; \underline{p}_I \qquad (3\text{-}72)$$

where

$\underline{p}_L$ is the target position vector in the LOS frame,

T_L^I is the Euler rotation matrix, Equation 3-41,

and

$\underline{p}_I$ is the target position vector, inertial frame.

2) Find the error angles between LOS and the target position vector:

$$e_a = \arctan\left(\underline{p}_{2L} / \underline{p}_{1L}\right) \qquad (3\text{-}73)$$

$$e_e = \arctan [\underline{P}_{3L} / (\underline{P}_{1L}^2 + \underline{P}_{2L}^2)^{(1/2)}] \tag{3-74}$$

where

e_a is the azimuth pointing error angle,

e_e is the elevation pointing error angle, and

$\underline{P}_{iL}$ is the component of the P_L vector in the i-direction, i=1,2,3.

3) Calculate the velocity of the target normal to the range vector by the cross product operation

$$\underline{R}_I = \underline{P}_I \times \underline{V}_I \tag{3-75}$$

where

$\underline{R}_I$ is the (estimated) target velocity normal to the range vector, expressed in the inertial frame,

$\underline{P}_I$ is the (estimated) inertial target position,

$\underline{V}_I$ is the (estimated) inertial velocity vector, and

X indicates the cross product operation.

4) Rotate the velocity normal to the range into the LOS frame, and convert to unit vectors

$$\underline{W}_L = T_L^I \ \underline{R}_I / \|\underline{R}_I\|^2 \tag{3-76}$$

where

$\underline{W}_L$ is the rate aiding vector in the LOS frame, and

$\|\underline{R}_I\|^2$ is the square of the relative target range.

5) Calculate the rate aiding terms as

$$W_n(1) = W_o(1) + 26.46 \cdot \Delta t \cdot e_a \tag{3-77}$$

$$W_n(2) = W_o(2) + 26.46 \cdot \Delta t \cdot e_e \tag{3-78}$$

where

$W_n(i)$ is the new rate aiding term in the i-direction, i=1,2,3,

$W_o(i)$ is the rate aiding term in the i-direction from the previous sample time (initialized to zero), and

Δt is the sample period (time between updates).

6) Calculate incremental gimbal rotations over the next propagation cycle as

$$\Delta a = (\underline{W}_L(3) + 12.6 \cdot e_a + W_n(1))\Delta t \tag{3-79}$$

$$\Delta e = (\underline{W}_L(2) + 12.6 \cdot e_e + W_n(2))\Delta t \tag{3-80}$$

where

Δa is the amount the gimbal rotates in azimuth in the next Δt time interval,

Δe is the amount the gimbal rotates in elevation in the next Δt time interval, and

$\underline{W}_L(i)$ is the i-th component of $\underline{W}_L$.

7) Calculate the Euler rotation matrix $T_{\Delta t}$ which rotates the sensor head to its desired position at the next sample time, based upon the incremental gimbal rotations from Equation 3-41.

8) Multiply the body-to-LOS Euler rotation matrix by the incremental gimbal rotation matrix to yield the body-to-LOS Euler rotation matrix for the next update:

$$T^B_{Ln} = T_{\Delta t} \, T^B_{Lo} \tag{3-81}$$

where

T^B_{Ln} is the body-to-LOS rotation matrix for the next update time,

$T_{\Delta t}$ is the incremental gimbal rotation matrix, and

T^B_{Lo} is the body-to-LOS rotation matrix for the present (just completed) update.

The transformation from inertial to LOS is given at the next update time by multiplying the inertial-to-body rotation matrix, obtained from attitude information from the INU, by the new body-to-LOS rotation matrix. Since a benign attacker is assumed in this thesis, the inertial-to-body transformation matrix (for this research only) is just a three-by-three identity matrix.

3.6. Coordinate Transformations

As discussed in Section 2.4.4, the covariances and states must be rotated into the LOS frame for update and back into the inertial frame for propagation. The states are rotated into the LOS frame by the transformation matrix given in Equation 3-41 before update:

$$\hat{x}_L^- = \begin{bmatrix} T_L^I & 0_{3X3} & 0_{3X3} \\ 0_{3X3} & T_L^I & 0_{3X3} \\ 0_{3X3} & 0_{3X3} & T_L^I \end{bmatrix} \hat{x}_I^- \qquad (3\text{-}82)$$

where

$\hat{x}_L^-$ is the state estimate before update at t_i in the LOS frame,

$\hat{x}_I^-$ is the state estimate before update at t_i in the inertial frame, and

T_L^I is the inertial-to-LOS transformation matrix computed by Equation (3-41).

Since the transformation matrix T_L^I used for this rotation is orthogonal, Equation (3-82) can be used to transform the states from the LOS frame to the inertial frame after update, if the transpose of T_L^I is used in place of T_L^I and the roles of $\hat{\underline{x}}_I$ and $\hat{\underline{x}}_L$ are interchanged. The covariances are rotated into the LOS frame in much the same manner, except a post-multiply by the transformation matrix is required:

$$P_L^- = \begin{bmatrix} T_L^I & 0_{3X3} & 0_{3X3} \\ 0_{3X3} & T_L^I & 0_{3X3} \\ 0_{3X3} & 0_{3X3} & T_L^I \end{bmatrix} P_I^- \begin{bmatrix} T_L^I & 0_{3X3} & 0_{3X3} \\ 0_{3X3} & T_L^I & 0_{3X3} \\ 0_{3X3} & 0_{3X3} & T_L^I \end{bmatrix}^T \qquad (3\text{-}83)$$

where

P_L^- is the covariance matrix before update at t_i in the LOS frame,

P_I^- is the covariance matrix before update at t_i in the inertial frame, and

T_L^I is the inertial-to-LOS transformation matrix computed by Equation (3-41).

3.7. Summary

The CTR filter is presented in this section. The filter dynamics equations, measurement equations, and noise models are presented, and implementation on SOFE is discussed. Finally, the method used in this thesis to rotate the states and covariances between the inertial and LOS coordinate frames is presented.

Because it models the real-world situation well, the CTR filter is used as a baseline in this thesis and few assumptions are made to simplify its complexity. The next chapter presents a simpler filter model and makes more simplifying assumptions to reduce recursion computation time. Results from both filter models are compared in Chapter V.

IV. Gauss-Markov Filter

4.1. Introduction

Two implementations of the Gauss Markov (GM) EKF are presented in this section. The first implementation uses the propagation and update techniques presented in Section 3.5 for the CTR filter. The second implementation uses the U-D covariance factorization techniques described in Section 2.4.3 for propagation and update. Both implementations are shown to be less complex than the CTR EKF presented in Chapter III. Reduced complexity is desirable because the TSE on board the AFTI/F-16 is limited in both memory allocation (32K of program space and 32K of dynamic RAM) and processing time (7 milliseconds per update frame) [10].

4.2. Filter Dynamics Model

The dynamics model of the GM filter uses the same states as the CTR filter, given in Equation (3-1). The dynamics model for the GM filter is

$$\dot{\hat{\underline{x}}} = \begin{bmatrix} \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_7 - a_1 \\ \hat{x}_8 - a_2 \\ \hat{x}_9 - a_3 \\ -\hat{x}_7/\tau_1 \\ -\hat{x}_8/\tau_2 \\ -\hat{x}_9/\tau_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} \qquad (4\text{-}1)$$

where

w_1, w_2, and w_3 are zero-mean, white Gaussian noises independent of each other and of $\hat{\mathbf{x}}$,

τ_1, τ_2, and τ_3 are correlation time constants for the acceleration states, and

a_1, a_2, and a_3 are the attacker accelerations in the 1, 2, and 3 directions.

The correlation time constants characterize the half-power point of the power spectral densities of the correlated acceleration. The frequency at which the half-power point occurs is assumed to be at about two Hertz for high-performance aircraft [8:61], yielding correlation time constants of two seconds. The correlation time constants are assumed to be equal in all three directions because the target is assumed to have the same acceleration capacity along any axis of the inertial frame [8:61].

Since the dynamics model is linear and time invariant, the dynamics matrix **F** of Equation (2-6) is calculated as

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_3 \end{bmatrix} \tag{4-2}$$

which is obviously a much simpler matrix to evaluate than the F matrix for the CTR filter (Equation (3-17)).

4.3. Measurement Equations

In the GM filter, the states and covariances are propagated and updated in the inertial frame, so the measurements are accomplished differently than in the CTR filter. The range and range rate measurement equations are are identical in any frame, and the covariances associated these measurements are equal in either the inertial or LOS frame. However, the covariances of the angle measurements are angles about the LOS vector, which are not the same in the different frames unless they describe a circular cone about the LOS vector. Unfortunately, the covariances describe a squared cone, as shown in Figure 5. The covariances are equal if the planes described by the 1-2 axes of the coordinate frames (the 'horizontal' planes) are aligned, and are skewed up to 45 degrees otherwise. The FLIR head is roll-stabilized in the aircraft [10], so the 'horizontal' planes should always be closely aligned.

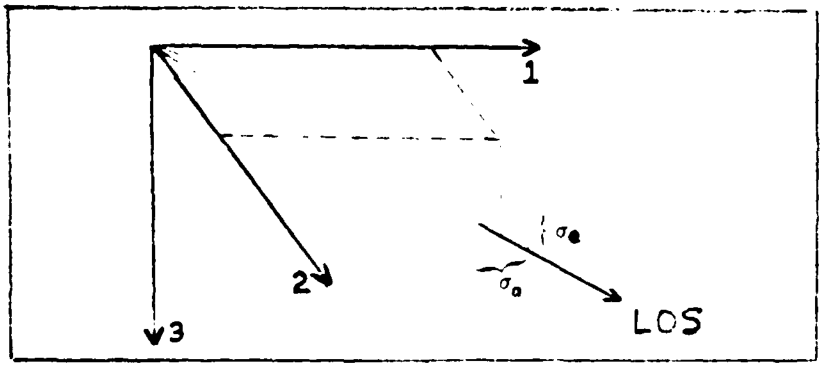


Figure 5. Angular Error Covariances

Since the states are not rotated into the LOS frame for update, different equations are used for the angular measurements: the angles are measured as Euler angles; i.e.,

$$\tan(a) = \hat{x}_2 / \hat{x}_1 \tag{4-3}$$

$$\tan(e) = \hat{x}_3 / (\hat{x}_1^2 + \hat{x}_2^2)^{(1/2)} \tag{4-4}$$

and the range and range rate measurements are given in Equations (3-36) and (3-40), respectively.

The H matrix is calculated as

$$H = \begin{bmatrix} H_1 & H_2 & H_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ H_4 & H_5 & H_6 & H_1 & H_2 & H_3 & 0 & 0 & 0 \\ H_7 & H_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ H_9 & H_{10} & H_{11} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4-5}$$

where

$$H_1 = \hat{x}_1 / R \tag{4-6}$$

$$H_2 = \hat{x}_2 / R \tag{4-7}$$

$$H_3 = \hat{x}_3 / R \tag{4-8}$$

$$H_4 = (\hat{x}_4 R^2 - \hat{x}_1(\hat{x}_1\hat{x}_4 + \hat{x}_2\hat{x}_5 + \hat{x}_2\hat{x}_6)) / R^3 \tag{4-9}$$

$$H_5 = (\hat{x}_5 R^2 - \hat{x}_2(\hat{x}_1\hat{x}_4 + \hat{x}_2\hat{x}_5 + \hat{x}_3\hat{x}_6)) / R^3 \tag{4-10}$$

$$H_6 = (\hat{x}_6 R^2 - \hat{x}_3(\hat{x}_1\hat{x}_4 + \hat{x}_2\hat{x}_5 + \hat{x}_3\hat{x}_6)) / R^3 \tag{4-11}$$

$$H_7 = -\hat{x}_2 / \hat{x}_1^2 \tag{4-12}$$

$$H_8 = 1 / \hat{x}_1 \tag{4-13}$$

$$H_9 = -\hat{x}_1\hat{x}_3 / (\hat{x}_1^2 + \hat{x}_2^2)^{3/2} \tag{4-14}$$

$$H_{10} = -\hat{x}_2\hat{x}_3 / (\hat{x}_1^2 + \hat{x}_2^2)^{3/2} \tag{4-15}$$

$$H_{11} = (\hat{x}_1^2 + \hat{x}_2^2)^{-(1/2)} \tag{4-16}$$

with

$$R = (\hat{x}_1^2 + \hat{x}_2^2 + \hat{x}_3^2)^{(1/2)} \qquad (4-17)$$

4.4. Noise Models

As with the CTR filter, two types of noise models are apparent in the GM filter: measurement and dynamics noises. Measurement noise models are identical for either filter because the same devices are used to perform the measurements. Dynamics noise models are acquired by assuming that zero-mean, white, Gaussian noises drive the acceleration states and performing a steady-state covariance analysis. According to Worsley [8:63], the results of this covariance analysis results in the equation

$$2\, p_{33}/\tau = q_1 \qquad (4-18)$$

where

p_{33} is the covariance associated with the acceleration state along any axis, and

q_1 is the strength of the driving noise.

Assuming the same value of error in the target acceleration estimate as in the CTR filter and the correlation time constant of 2 seconds, the value of q_i, i=1,2,3 is 21000 feet^2 per second^5, the same as for the CTR filter.

4.5. Acceleration Rotation

The CTR filter dynamics equations use a constant target turn rate model, as developed in Appendix B, to correlate the acceleration states with the velocity states in an effort to match the expected target performance in an aerial engagement [8:3,22]. The GM filter dynamics model does not correlate the acceleration states to anything, but rather assumes that the target is always capable of equal accelerations in all directions. Thus, the GM filter does not effectively model a real aircraft, because most aircraft accelerations occur normal to the aircraft's flight path. One method of forcing the GM filter to model a real aircraft more closely is to rotate the acceleration states to make them perpendicular to the velocity states just before the states are updated [10]. The acceleration rotation is accomplished using two vector cross products, as shown below:

$$\underline{A}_n = \underline{V}_T \text{ X } (\underline{A}_o \text{ X } \underline{V}_T) \tag{4-19}$$

where

$\underline{A}_n$ is the new (rotated) acceleration vector,

$\underline{A}_o$ is the old (non-rotated) acceleration vector,

$\underline{V}_T$ is the target total velocity vector, and

X denotes the cross product operation.

For both implementations of the GM filter, the rotation above is performed on the state estimates after propagation but before update. Thus, the GM filters used in this thesis are modified GM filters.

4.6. GM Filter Implementation

As mentioned in Section 4.1, the GM filter is implemented two ways in this thesis. The first implementation mirrors the CTR filter implementation, shown in Section 3.5, using Carlson square root covariance updates and propagation by integration. The second implementation uses *U-D covariance factorization* techniques to update and propagate the covariance matrix, and the state transition matrix to propagate the filter states. An overview of these techniques is presented in Section 2.4.3; the remainder of this section is devoted to explaining the implementation of these techniques using SOFE.

The state transition matrix, $\Phi(t,t_o)$, propagates the states from t_o to t and is used to propagate the covariances. For a linear, time-invariant system dynamics model such as the GM filter, the state transition matrix can be calculated as an inverse Laplace transform [4:42]:

$$\Phi(t-t_o) = \mathcal{L}^{-1}\{[sI - F]^{-1}\} \tag{4-19}$$

where

$\mathcal{L}^{-1}$ is the inverse Laplace transform operator, and
s is the Laplace integration variable.

The state transition matrix for the GM filter is calculated (assuming $\tau_1=\tau_2=\tau_3=\tau$) as

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & t & 0 & 0 & \Phi_1 & 0 & 0 \\ 0 & 1 & 0 & 0 & t & 0 & 0 & \Phi_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & t & 0 & 0 & \Phi_1 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Phi_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Phi_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Phi_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Phi_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Phi_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Phi_3 \end{bmatrix} \tag{4-21}$$

where

$$\Phi_1 = \tau \Delta t - \tau^2 + \tau^2 \exp(-\Delta t/\tau) \qquad (4\text{-}22)$$

$$\Phi_2 = \tau - \tau \exp(-\Delta t/\tau) \qquad (4\text{-}23)$$

$$\Phi_3 = \exp(-\Delta t/\tau) \qquad (4\text{-}24)$$

Since Equations (4-22) and (4-23) are differences of numbers of similar magnitude, their calculation may be inaccurate if done by a small wordlength computer, especially for small values of Δt. For this reason, Equations (4-22) to (4-24) are expanded in a Taylor's series. The series is trucated after the first two terms, resulting in the approximations

$$\Phi_1 = \sum_{n=2}^{\infty} (-1)^n \frac{\Delta t^n}{n!\ \tau^{n-1}} \approx \Delta t^2/2 - \Delta t^3/(6\tau) \qquad (4\text{-}25)$$

$$\Phi_2 = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{\Delta t^n}{n!\ \tau^{n-1}} \approx \Delta t - \Delta t^2/(2\tau) \qquad (4\text{-}26)$$

$$\Phi_3 = \sum_{n=0}^{\infty} (-1)^n \frac{\Delta t^n}{n!\ \tau^n} \approx 1 - \Delta t/\tau \qquad (4\text{-}27)$$

These approximations introduce a maximum error of 0.0014 percent into the calculations for Φ.

To propagate the states if an input (attacker acceleration, see Equation (4-1)) were present, the B_d matrix of Equation (2-23), as defined in Equation (2-24), is required [4:171]:

$$B_d = \int_{t_i}^{t_{i+1}} \Phi(t_{i+1},\tau)\ B(\tau)\ d\tau \qquad (4\text{-}28)$$

where B is the continuous-time input matrix imbedded in Equation (4-1). Assuming the input is the three-element attacker acceleration vector, the discrete-time input matrix is calculated as

$$B_d = \begin{bmatrix} \Delta t^2/2 & 0 & 0 \\ 0 & \Delta t^2/2 & 0 \\ 0 & 0 & \Delta t^2/2 \\ -\Delta t & 0 & 0 \\ 0 & -\Delta t & 0 \\ 0 & 0 & -\Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4-29}$$

where $\Delta t = t_{i+1} - t_i$

The covariances are propagated by using Equations (2-18) to (2-22). The values for G_d and Q_d in these equations is calculated by assuming that G, the continuous-time input matrix, is a nine-by-nine identity matrix and Q, the continuous-time noise matrix, is given by Equation (3-70). Then, Equation (2-20) is used to calculate Q_d [4:171] as

$$Q_d = \begin{bmatrix} Q_1 & 0 & 0 & Q_2 & 0 & 0 & Q_3 & 0 & 0 \\ 0 & Q_1 & 0 & 0 & Q_2 & 0 & 0 & Q_3 & 0 \\ 0 & 0 & Q_1 & 0 & 0 & Q_2 & 0 & 0 & Q_3 \\ Q_2 & 0 & 0 & Q_4 & 0 & 0 & Q_5 & 0 & 0 \\ 0 & Q_2 & 0 & 0 & Q_4 & 0 & 0 & Q_5 & 0 \\ 0 & 0 & Q_2 & 0 & 0 & Q_4 & 0 & 0 & Q_5 \\ Q_3 & 0 & 0 & Q_5 & 0 & 0 & Q_6 & 0 & 0 \\ 0 & Q_3 & 0 & 0 & Q_5 & 0 & 0 & Q_6 & 0 \\ 0 & 0 & Q_3 & 0 & 0 & Q_5 & 0 & 0 & Q_6 \end{bmatrix} \qquad (4\text{-}30)$$

where, using the same expansions as for Equation (4-21),

$$Q_1 = q \sum_{n=5}^{\infty} (-1)^n \frac{2(n!) - 2^{n-1}(n-1)!}{n!\,(n-1)!} \frac{\Delta t^n}{\tau^{n-1}}$$

$$\approx q[\Delta t^5/20 - \Delta t^6/(36\tau)] \qquad (4\text{-}31)$$

$$Q_2 = q \sum_{n=4}^{\infty} (-1)^n \frac{(2^{n-1}-1)(n-1)! - n!}{n!\,(n-1)!} \frac{\Delta t^n}{\tau^{n-1}}$$

$$\approx q[\Delta t^4/8 - \Delta t^5/(12\tau)] \qquad (4\text{-}32)$$

$$Q_3 = q \sum_{n=3}^{\infty} (-1)^n \frac{n! - 2^{n-1}(n-1)!}{n!\,(n-1)!} \frac{\Delta t^n}{\tau^{n-1}}$$

$$\approx q[\Delta t^3/6 - \Delta t^4/(6\tau)] \qquad (4\text{-}33)$$

$$Q_4 = q \sum_{n=3}^{\infty} (-1)^n \frac{2 - 2^{n-1}}{n!} \frac{\Delta t^n}{\tau^{n-1}}$$

$$\approx q[\Delta t^3/3 - t^4/(4\tau)] \qquad (4\text{-}34)$$

$$Q_5 = q \sum_{n=2}^{\infty} (-1)^n \frac{2^{n-1}-1}{n!} \frac{\Delta t^n}{\tau^{n-1}}$$

$$\approx q[\Delta t^2/2 - \Delta t^3/(2\tau)] \qquad (4\text{-}35)$$

$$Q_6 = q \sum_{n=1}^{\infty} (-1)^{n-1} \frac{2^n}{n!} \frac{\Delta t^n}{\tau^{n-1}}$$

$$\approx 2q[\Delta t - \Delta t^2/\tau] \qquad (4\text{-}36)$$

The errors incurred by truncating the series after the first two terms are less than 0.0094 percent of each Q value.

In the equations for propagating the covariances (Equations (2-18) to (2-22)), the Q_d matrix is assumed diagonal and the G_d matrix is upper triangular. The required forms of these matrices are acquired by breaking the Q_d matrix of Equation (4-30) into an upper diagonal unitary matrix and a diagonal matrix using the algorithm presented in Equations (2-13) and (2-14). Note that this is the same algorithm used to generate the U and D factors of the initial covariance matrix, so additional programming is not required.

To implement the second form of the GM filter in SOFE, several modifications are required. First, the integration routines are removed and replaced with the propagation algorithms presented in Section 2.4.3. Then, the Carlson square root update algorithms are replaced with the U-D update algorithms, also presented in Section 2.4.3. The filter dynamics models are removed, because these models are imbedded in the propagation algorithms. Much of the remainder of the program is then modified to allow several input/output calls, embedded in the removed routines, to be performed.

4.7. Summary

The GM EKF is presented in this chapter. The GM dynamics model is shown to be less complex than the CTR filter dynamics model of the previous chapter. Also, by not rotating the states and modifying the measurements, the filter becomes less complex to simulate because the sensor head need not be manipulated in order to acquire the inertial-to-LOS transformation matrix.

In Chapter V, results of the GM EKFs are compared to the results from the CTR filter of Chapter III.

V. Results and Conclusions

5.1. Introduction

This chapter presents the results of the SOFE simulations of the filters developed in Chapters III and IV. Results from each filter are addressed, then the tracking capabilities of all the filters are compared. Finally, conclusions of the research and recommendations for further study are presented.

Each SOFE simulation exercised a filter over 20 Monte Carlo runs to monitor the statistical performance of the filter. Each of these runs is made against a file of trajectory data that is invariant from run to run and filter to filter. In this way, the relative performance of the various filters can be compared.

5.2. Trajectory Generation

Trajectories for the simulations are generated by computing the effects of an acceleration on a point mass traveling at a specified speed. Any changes in applied accelerations were accomplished at the rate of nine g's per second. Only basic planar moves were attempted; the point mass first executed a nine g right turn at three seconds into the simulation, then a nine g left turn at five seconds. At eight seconds, the horizontal accelerations dropped (over a one-second interval) to zero. Then, a similar move was made in the vertical plane; first, a 5 g downward turn at 9 seconds, then a 5 g upward turn at 11 seconds, and falling off to 0 at 14 seconds. The target is thus propagated forward from a given set of initial conditions and compared to the position of the attacker, which is always headed due north at 1000 feet per second. The relative position, relative velocity, and total target position, all referenced to the inertial coordinate frame, are then stored in a trajectory file.

Two sets of initial conditions generated two different trajectories. The first set of initial conditions places the target 5000 feet directly ahead of the attacker, going in the same direction at the same speed, as depicted in Figure VI. The second set of initial conditions places the target 10,000 feet away at a -45 degree angle from the heading of the attacker (to the attacker's left). In the second trajectory, the target is travelling at 1100 feet per second and at a 45 degree heading; i.e., the target is crossing in front of the attacker, and the first accelerations are applied towards the attacker (see Figure VII).

The trajectories are computed at a 3000 Hertz rate to ensure high accuracy of the simulation results. However, data are output to the trajectory file at a 30 Hertz (simulated) rate to be compatible with the filter sample rate. If the data rate of the trajectory file is different from the system sample rate, SOFE interpolates the trajectory data as needed using cubic splines. This interpolation requires significant processing time each cycle (frame) [6].

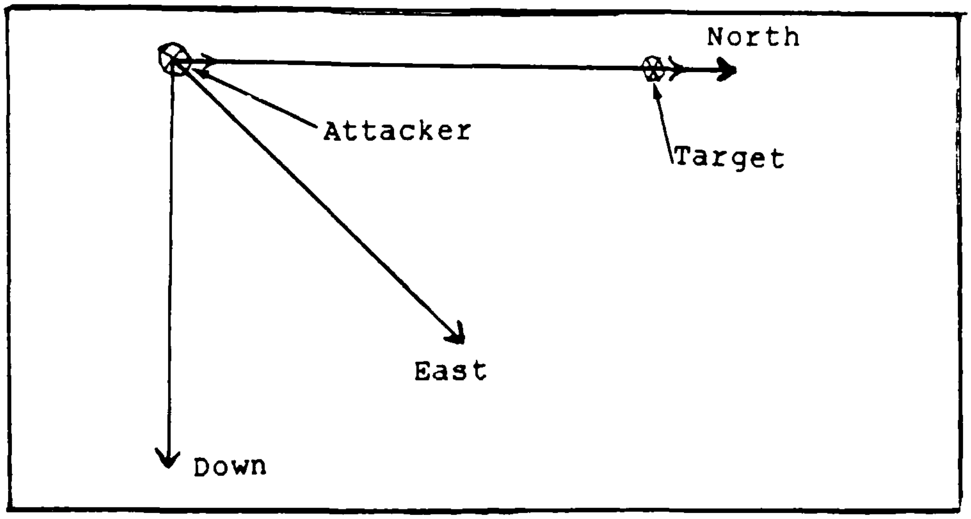


Figure 6. Trajectory 1 Initial Conditions

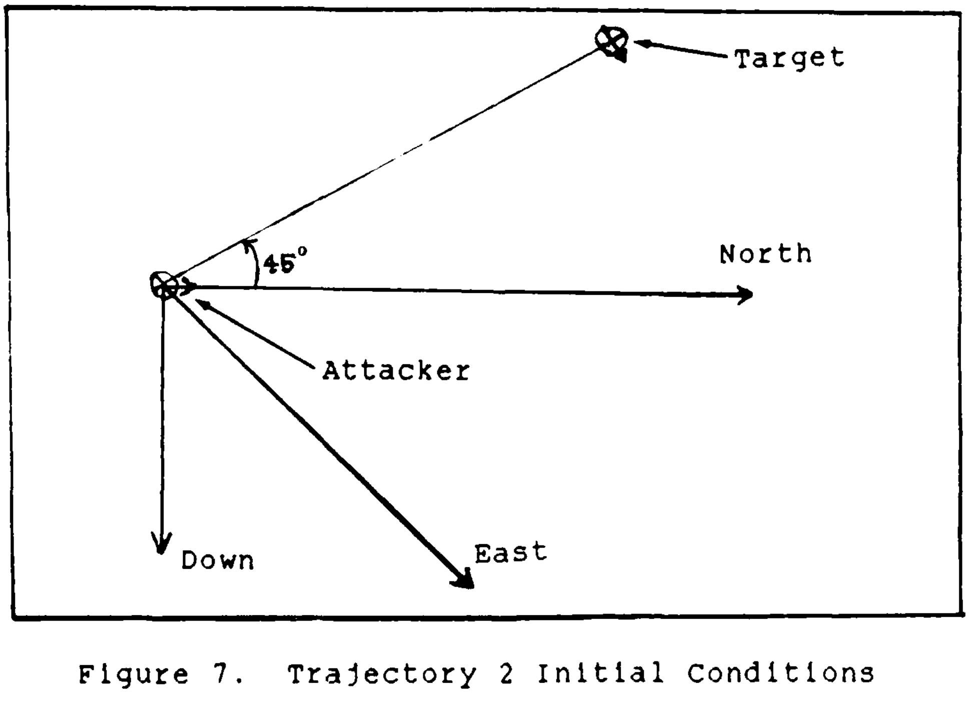


Figure 7. Trajectory 2 Initial Conditions

5.3. Filter Results

Each filter is run on SOFE for 20 Monte Carlo runs for each trajectory file. Twenty Monte Carlo runs is considered to be adequate based upon an analysis done by Worsley [8]. For each filter/trajectory combination, an ensemble average across all 20 runs is obtained. Plots of these ensemble averages are presented in Appendix C. The plots reflect the error between the filter estimated state and the trajectory state, and the covariance associated with each state, one state per plot. An analysis of the plots follows; Table II summarizes the maximum position error along each axis and the maximum total position error for each of the filter/trajectory combinations.

The CTR filter keeps the error between the estimated state and the trajectory data well within the covariance boundaries for all but a short time during the simulation. Also, the position states always have less than three feet total error along any axis. The CTR filter appears to be tracking well.

Two different results are obtained for the two implementations of the GM filter. The filter which integrates the equations forward in time had results nearly identical to those of the CTR filter. However, the GM filter which used propagation and U-D covariance factorization techniques tracked comparatively poorly; the maximum total errors are as high as 20 feet at times.

Table II

Filter Results

Filter	Trajectory	Maximum Error (Feet)			
		1-axis	2-axis	3-axis	Total(RMS)
CTR	1	1.3	2.7	2.7	2.8
GM	1	0.7	3.4	2.6	3.5
GM(UD)	1	5.3	19.5	12.5	20.3
CTR	2	1.3	2.7	2.7	2.7
GM	2	1.1	0.9	4.9	5.0
GM(UD)	2	3.5	17.8	14.0	17.8

5.4. Conclusions

Apparently, a flaw exists in the implementation of the U-D factorization GM filter. The U-D filter is tracking, but not quite as well as expected, given the performance of the other two filters. Varying system parameters seems to make performance worse rather than better. Testing of the various routines reveals that the flaw is probably in the routines which propagate the states and covariances forward in time, and not in the update routines. A possibility exists that SOFE does not work properly without some of the routines which are removed for this filter implementation.

Not rotating the states and covariances into the LOS frame for update seems to have very little effect on filter performance. The CTR filter, which did the rotation, performs little better than the integrated-forward GM filter, which does not do the rotation.

5.5. Recommendations

Each of the filters implemented in this thesis uses numerically stable covariance update techniques. Two quite different dynamics models are used with comparable results, and the covariance matrix rotation during update has no significant effect on filter performance. However, the previous AFTI/F-16 TSE does not perform as desired [2]. Since the filter dynamics models and covariance rotations had no significant effect, the effect of using conventional covariance update techniques should be investigated.

Also, the effects of finite wordlength should be investigated in more detail. The equations developed in this thesis can be run on a comparatively short wordlength computer, such as a Zenith Z-248, to study the effects of truncated wordlengths, or the wordlength of each quantity can be trucated after each operation to simulate the shorter wordlength.

Appendix A: Euler Angle Development

This appendix briefly presents the Euler angle relationships used in this thesis. Three reference frames are used: the inertial frame, which is stationary for all time, the (attacker aircraft) body axis, which may both translate and rotate, and the tracker (or LOS) frame.

The first reference frame, the inertial frame, is defined by the orientation of the INU and denoted by a subscript 'I'. The origin of the axes is defined as the center of the INU, because all rotations are measured around that point. A common name for this type of inertial frame is a wander azimuth frame, because the 'down' direction is always defined and 'ahead' or 'right' can be defined in any convenient manner.

The second reference frame, the body frame, is defined to have the 1-axis pointed out the nose of the aircraft, the 2-axis pointed along the right wing, and the 3-axis pointed out the 'bottom' of the aircraft body. The body frame, denoted by the subscript 'B', has its center at the center of gravity of the aircraft.

The third reference frame, the LOS frame, is defined by the orientation of the sensor head. The 1-axis points perpendicular to the plane of the sensor (FLIR) array and the 3-axis points 'down' (the sensor head is roll-stabilized). The 2-axis completes the right-handed coordinate frame (it points to the 'right').

Each reference frame can be related to the next by Euler angle rotations. These rotations are defined by two angles, say θ and ϕ. These angles define how much the base reference frame is rotated, first to the right in azimuth then down in elevation, to match the orientation of the second reference frame.

Although Euler angles are easily defined for most relative orientations, they are ill-conditioned if the second Euler angle approaches 90 degrees. If the second Euler angle equals 90 degrees, the first Euler angle is undefined because the 1-axis of the second reference frame cannot be projected into the 1-2 plane of the base reference frame. The possibility of this occurring is assumed to be negligible in this thesis; other coordinate transformations, such as direction cosines or quaternions, could be used to eliminate any problem with reference frame orientations.

However, because Euler rotations are performed, especially in the INU where the STS has no control, one property of Euler rotations is especially important in this thesis. This property is the following: if two or more Euler rotations are performed to go from the basis reference frame to the measurement reference frame, the orientation of the measurement reference frame with respect to the basis reference frame can <u>not</u>, in general, be described by a single Euler rotation using the sums of the rotation angles. Therefore, because the angle measurements are performed in the sensor head frame, the position of the sensor head must be known (or estimated) in the Kalman filter. Estimated angles are then rotated into the sensor head reference frame and compared to the actual measurement. Conversely, the measurement could be rotated into the basis (inertial) frame and compared to the estimated line-of-sight; this thesis uses the former approach because the covariance matrix undergoes the same rotations during measurement update (as explained in Chapter II).

Appendix B. Constant Turn Rate Coefficient

The coefficient $\|\omega\|^2$ used in the constant turn rate (CTR) filter is derived in Reference 8 and included here to make this thesis as complete a work as possible.

The CTR coefficient $\|\omega\|^2$, which is the square of the magnitude of the target's inertial turn rate, is developed from the application of the Coriolis theorem, written as

$$\frac{{}^{I}d}{dt}\underline{V}_T = \frac{{}^{T}d}{dt}\underline{V}_T + \underline{\omega}^I \times \underline{V}_T \qquad \text{(B-1)}$$

where

$\underline{V}_T$ is the inertial target velocity,

$\underline{\omega}^I$ is the inertial target angular velocity,

X denotes the cross-product operation, and

the superscripts I and T before the derivatives indicate that the derivatives are taken in the inertial and target body frame, respectively.

Now, the first term of the right-hand side of Equation (B-1) is zero because the target is assumed to be at constant speed. Taking the derivative of the remainder of Equation (B-1) with respect to time yields

$$\frac{{}^{I}d^2}{dt^2}\underline{V}_T = \frac{{}^{I}d}{dt}(\underline{\omega}^I \times \underline{V}_T) \qquad \text{(B-2)}$$

or, expressed in the target's body frame,

$$\frac{{}^{I}d^2}{dt^2}\underline{V}_T = \frac{{}^{T}d}{dt}(\underline{\omega}^I \times \underline{V}_T) + (\underline{\omega}^I \times (\underline{\omega}^I \times \underline{V}_T)) \qquad \text{(B-3)}$$

Now, since both the target speed and angular velocity are assumed constant, the first term on the right-hand side of Equation (B-3) is zero, yielding

$$\frac{{}^{I}d^{2}}{dt^{2}} \underline{V}_T = \underline{\omega}^I \times (\underline{\omega}^I \times \underline{V}_T) \qquad \text{(B-4)}$$

Using the relationship for a triple cross product, Equation (B-4) is written as

$$\frac{{}^{I}d^{2}}{dt^{2}} \underline{V}_T = (\underline{\omega}^I \bullet \underline{V}_T)\, \underline{\omega}^I - (\underline{\omega}^I \bullet \underline{\omega}^I)\, \underline{V}_T \qquad \text{(B-5)}$$

The first term of Equation (B-5) is zero since, for a planar turn, the target's inertial velocity and angular velocity vectors are perpendicular. Thus, Equation (B-5) becomes

$$\frac{{}^{I}d^{2}}{dt^{2}} \underline{V}_T = -\, \|\underline{\omega}\|^2\, \underline{V}_T \qquad \text{(B-6)}$$

which is the vector form of the derivatives of the acceleration states of the CTR filter.

To compute $\|\underline{\omega}\|^2$, the target's inertial acceleration vector, $\underline{A}_T$, is written as

$$\underline{A}^T = \underline{\omega}^I \times \underline{V}_T \qquad \text{(B-7)}$$

for a target flying at a constant speed. Taking the cross-product of the target's velocity vector into both sides of Equation (B-7) yields

$$\underline{V}_T \times \underline{A}_T = \underline{V}_T \times (\underline{\omega}^I \times \underline{V}_T) \qquad \text{(B-8)}$$

or, using the triple cross product relationship,

$$\underline{V}_T \times \underline{A}_T = (\underline{V}_T \bullet \underline{V}_T)\,\underline{\boldsymbol{\omega}}^I - (\underline{V}_T \bullet \underline{\boldsymbol{\omega}}^I)\,\underline{V}_T \tag{B-9}$$

Again, since the target's inertial velocity and angular velocity vectors are perpendicular, the last term on the right-hand side is zero. Rearranging Equation (B-9) gives

$$\underline{\boldsymbol{\omega}}^I = (\underline{V}_T \times \underline{A}_T) \,/\, \|\underline{V}_T\|^2 \tag{B-10}$$

Since

$$\|\underline{\boldsymbol{\omega}}^I\|^2 = (\underline{\boldsymbol{\omega}}^I \bullet \underline{\boldsymbol{\omega}}^I) \tag{B-11}$$

substitution of Equation (B-10) into Equation (B-11) yields

$$\|\underline{\boldsymbol{\omega}}^I\|^2 = [(\underline{V}_T \times \underline{A}_T)\bullet(\underline{V}_T \times \underline{A}_T)] \,/\, \|\underline{V}_T\|^4 \tag{B-12}$$

In the thesis, then, the appropriate state estimates replace their vector representations in Equation (B-12).

Appendix C. Plotted Data

The following are plots of the ensemble averages of data for each of the filters evaluated against each of the trajectories. Six filter/trajectory combinations are plotted, in plot sections numbered one through six. The first three plot sections are for each filter evaluated against Trajectory 1, as defined in Section 5.2. The last three plot sections are for Trajectory 2. Each plot section is subdivided into plots a through i, representing filter states x_1 through x_9, respectively. Thus, the plotted data is readily compared across filter types and trajectories by comparing plots with the same letter designator.

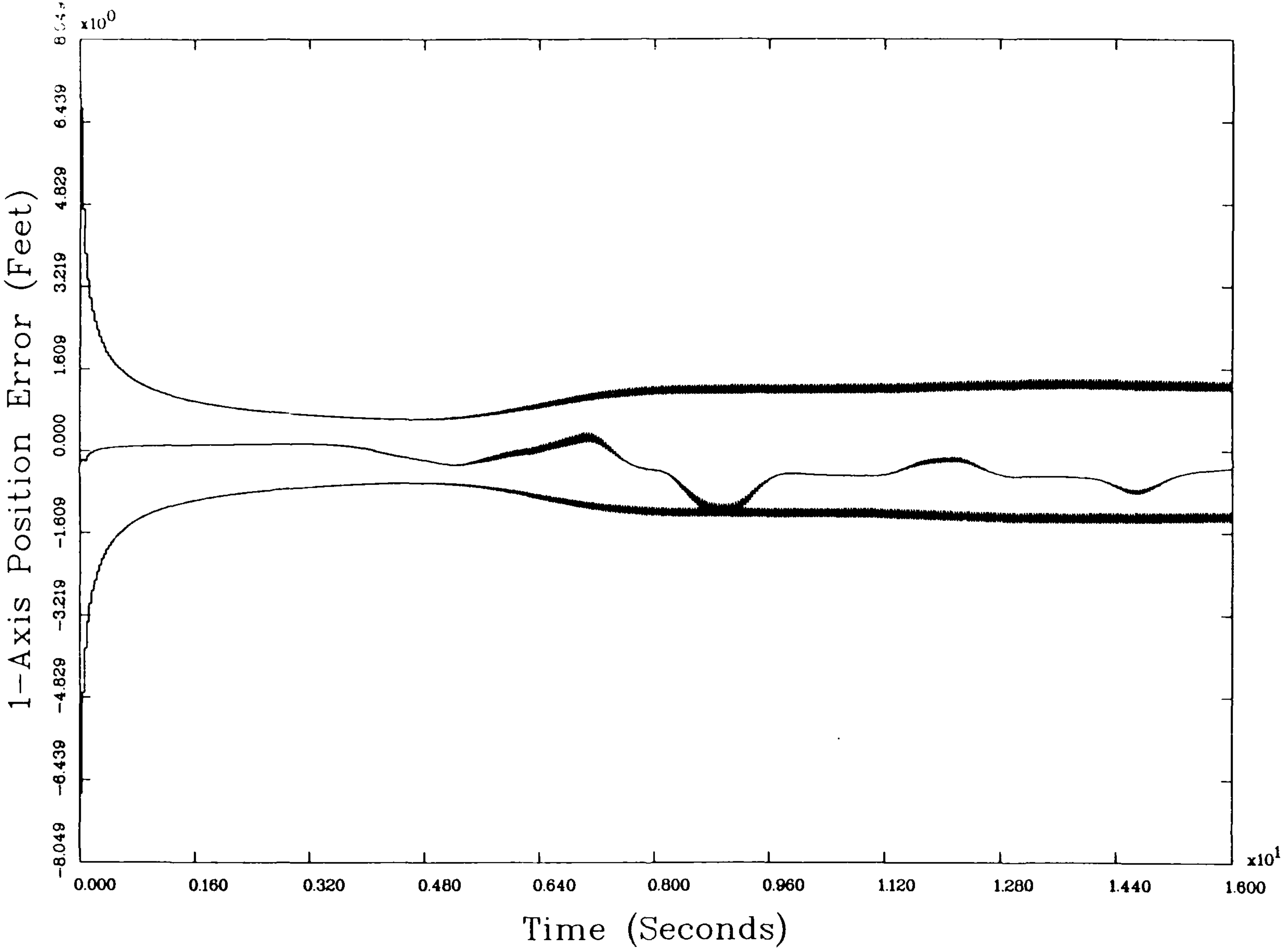


Figure 8.a. CTR Filter, Trajectory 1, x_1

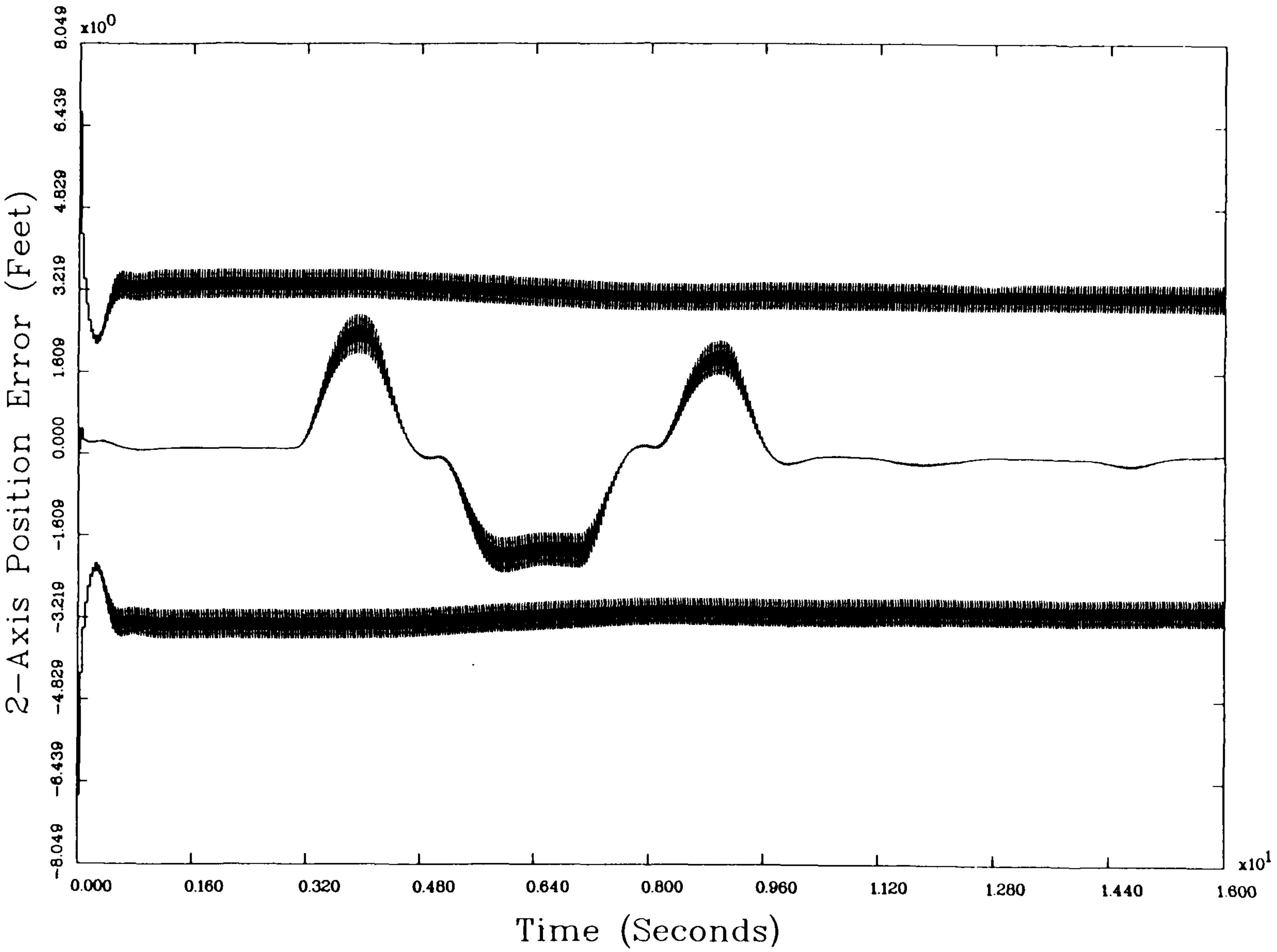


Figure 8.b. CTR Filter, Trajectory 1, x_2

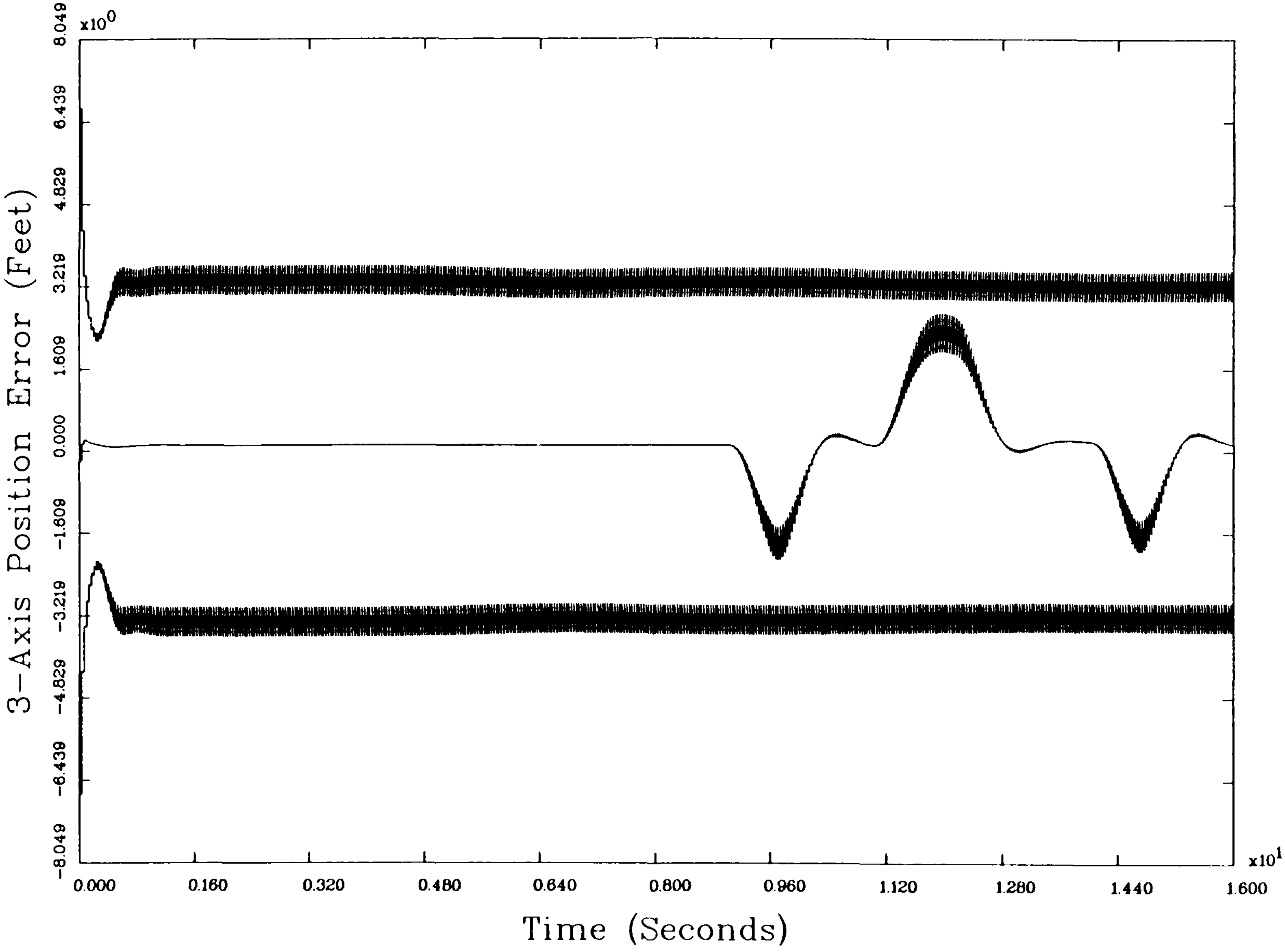


Figure 8.c. CTR Filter, Trajectory 1, x_3

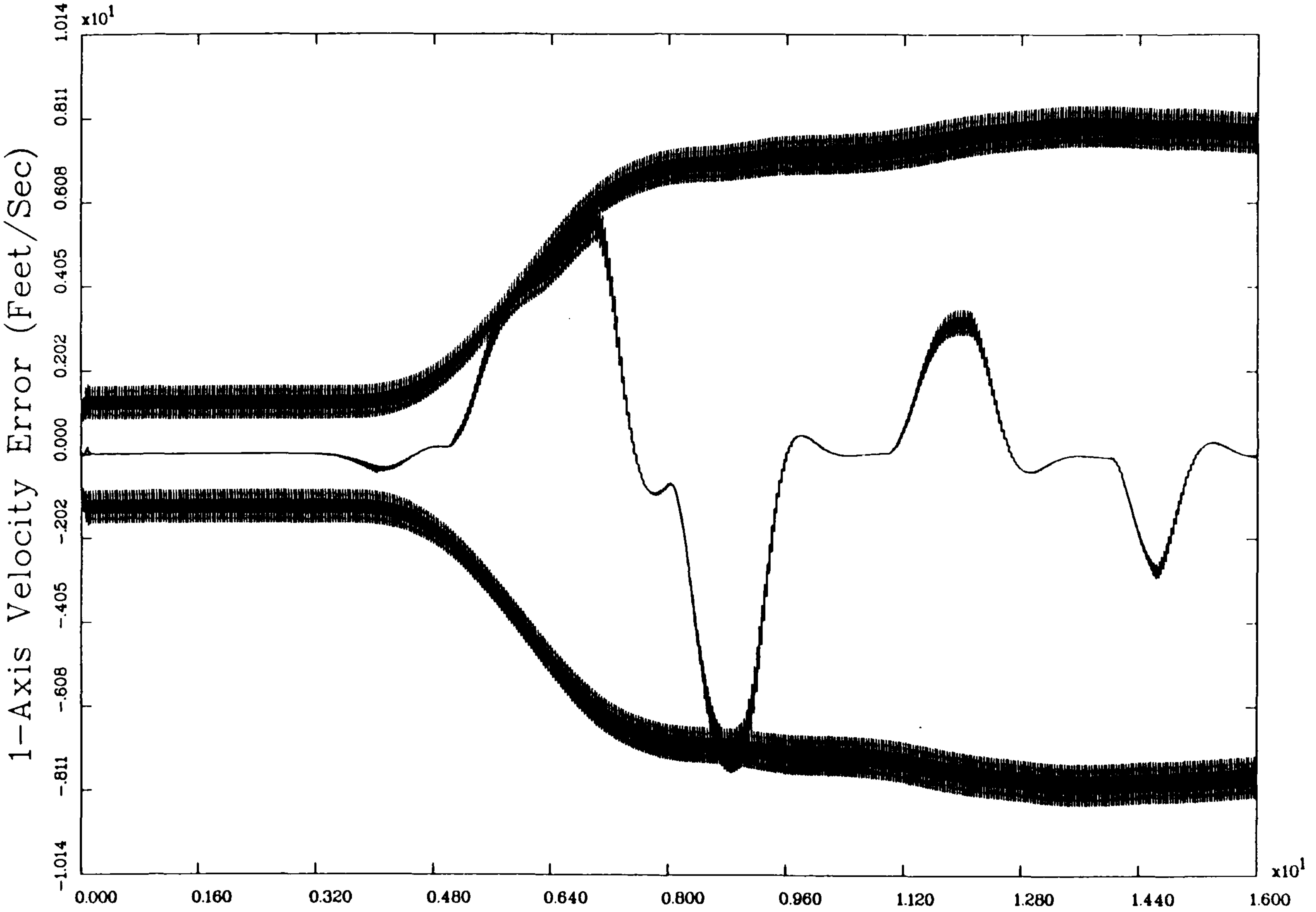


Figure 8.d. CTR Filter, Trajectory 1, x_4

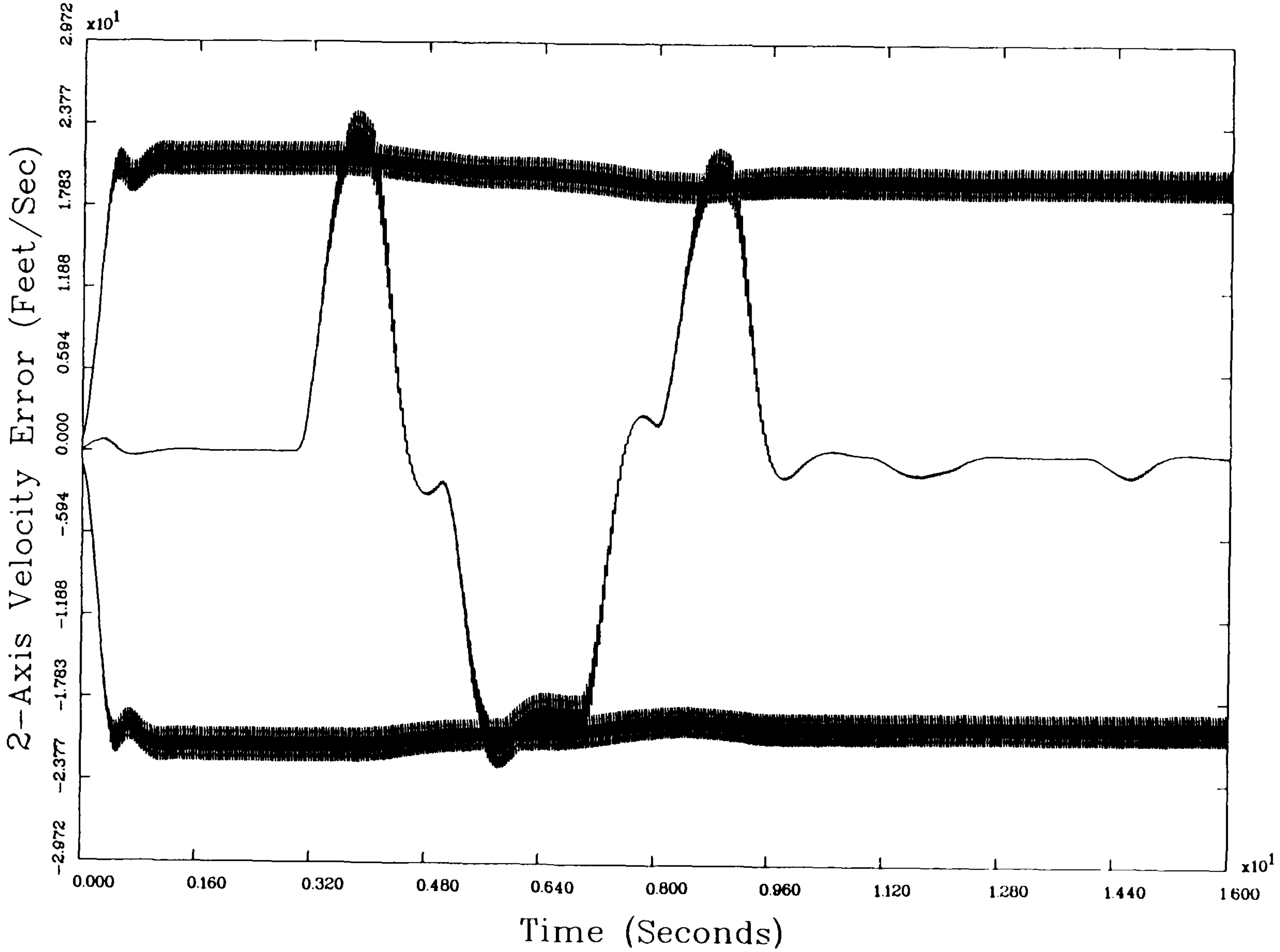


Figure 8.e. CTR Filter, Trajectory 1, x_5

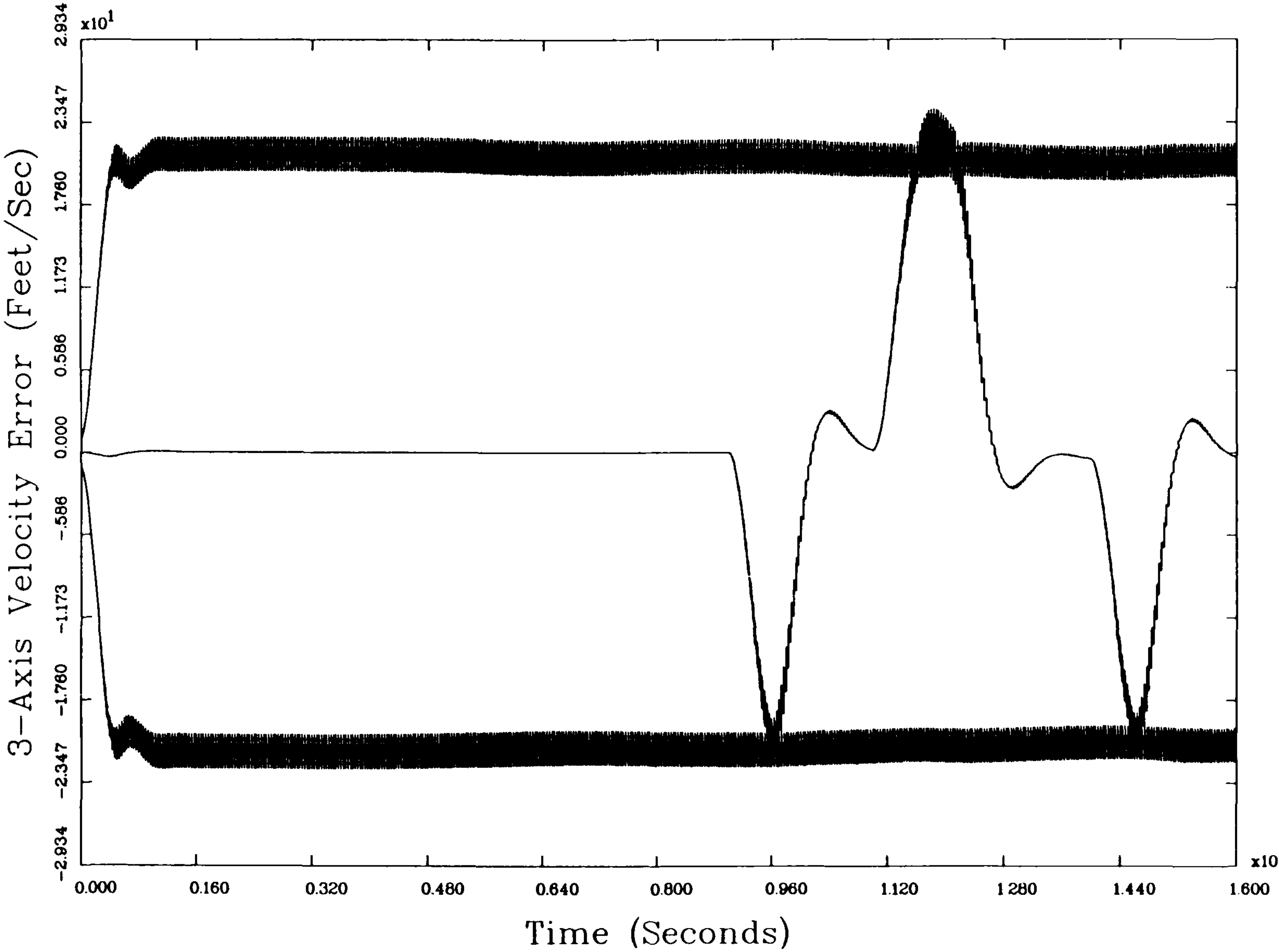


Figure 8.f. CTR Filter, Trajectory 1, x_6

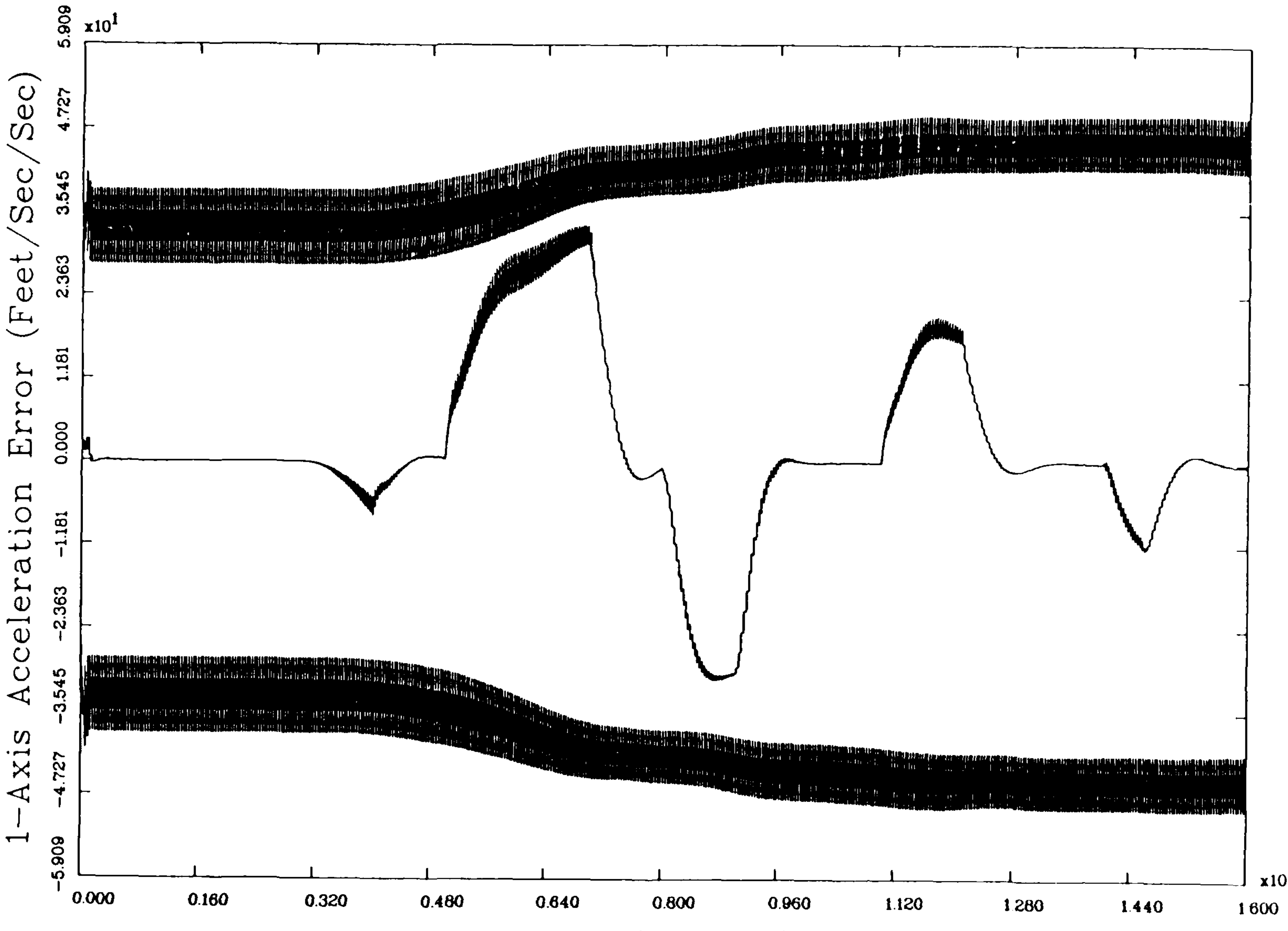


Figure 8.g. CTR Filter, Trajectory 1, x_7

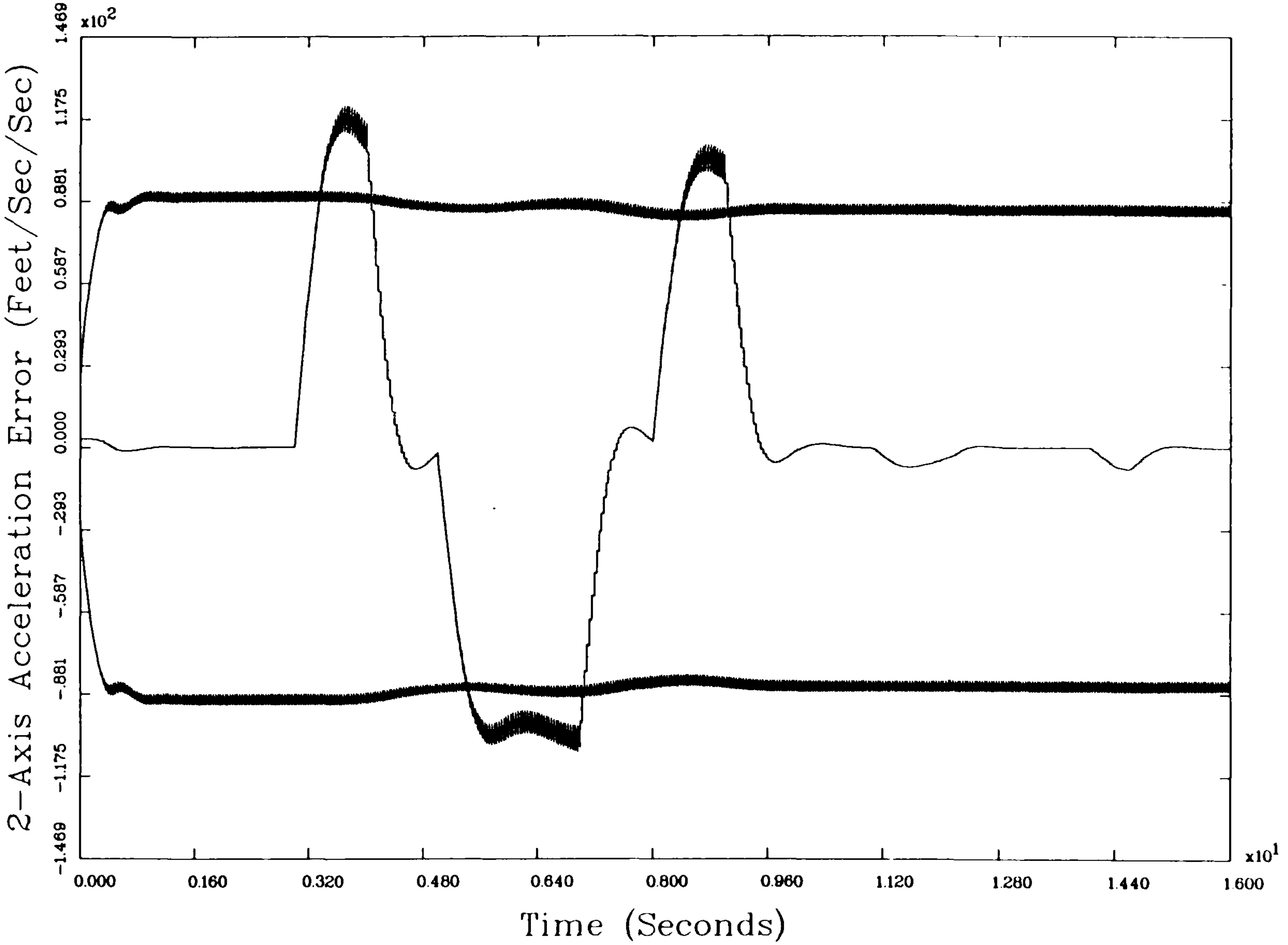


Figure 8.h. CTR Filter, Trajectory 1, x_8

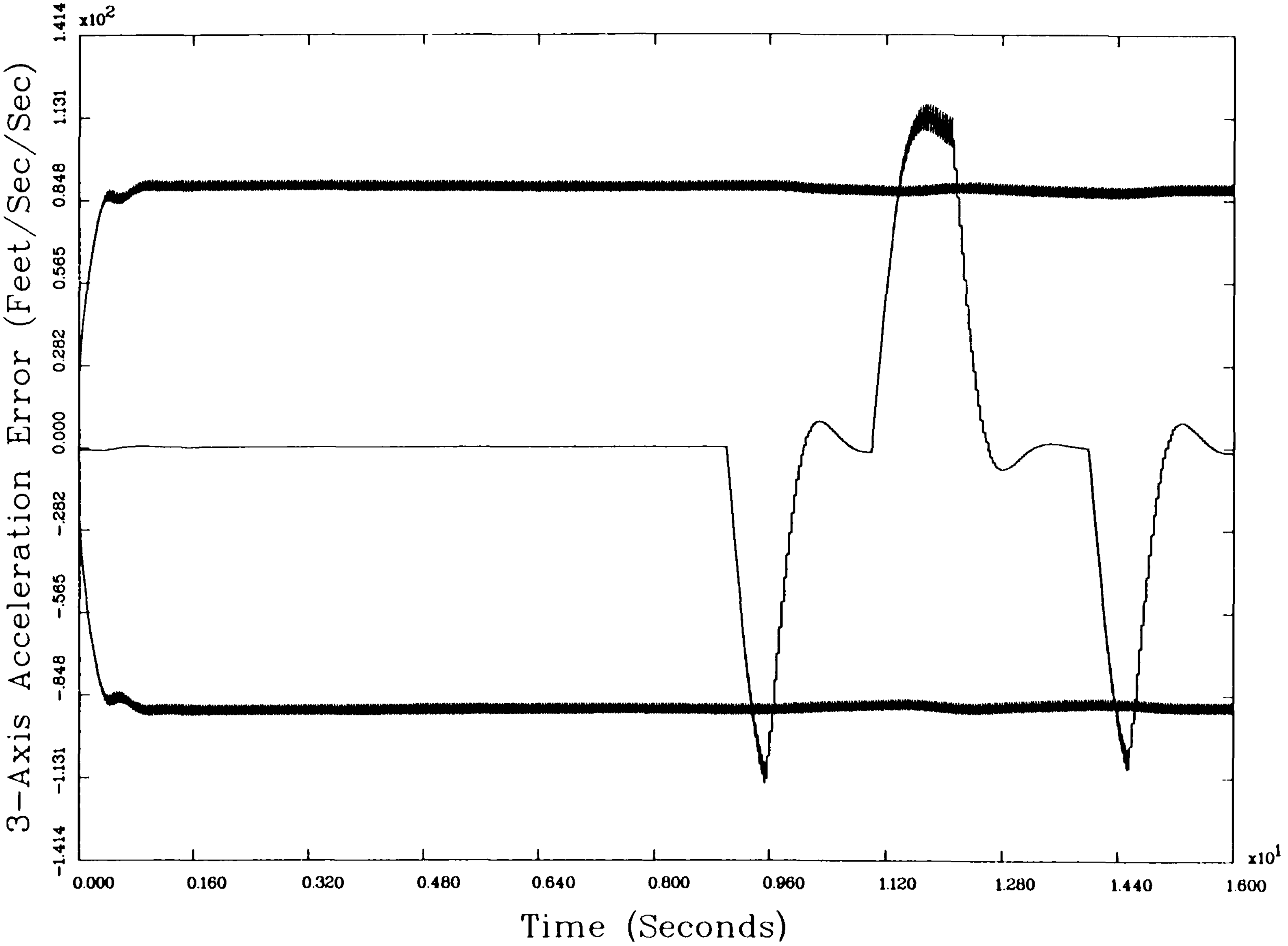


Figure 8.1. CTR Filter, Trajectory 1, x_9

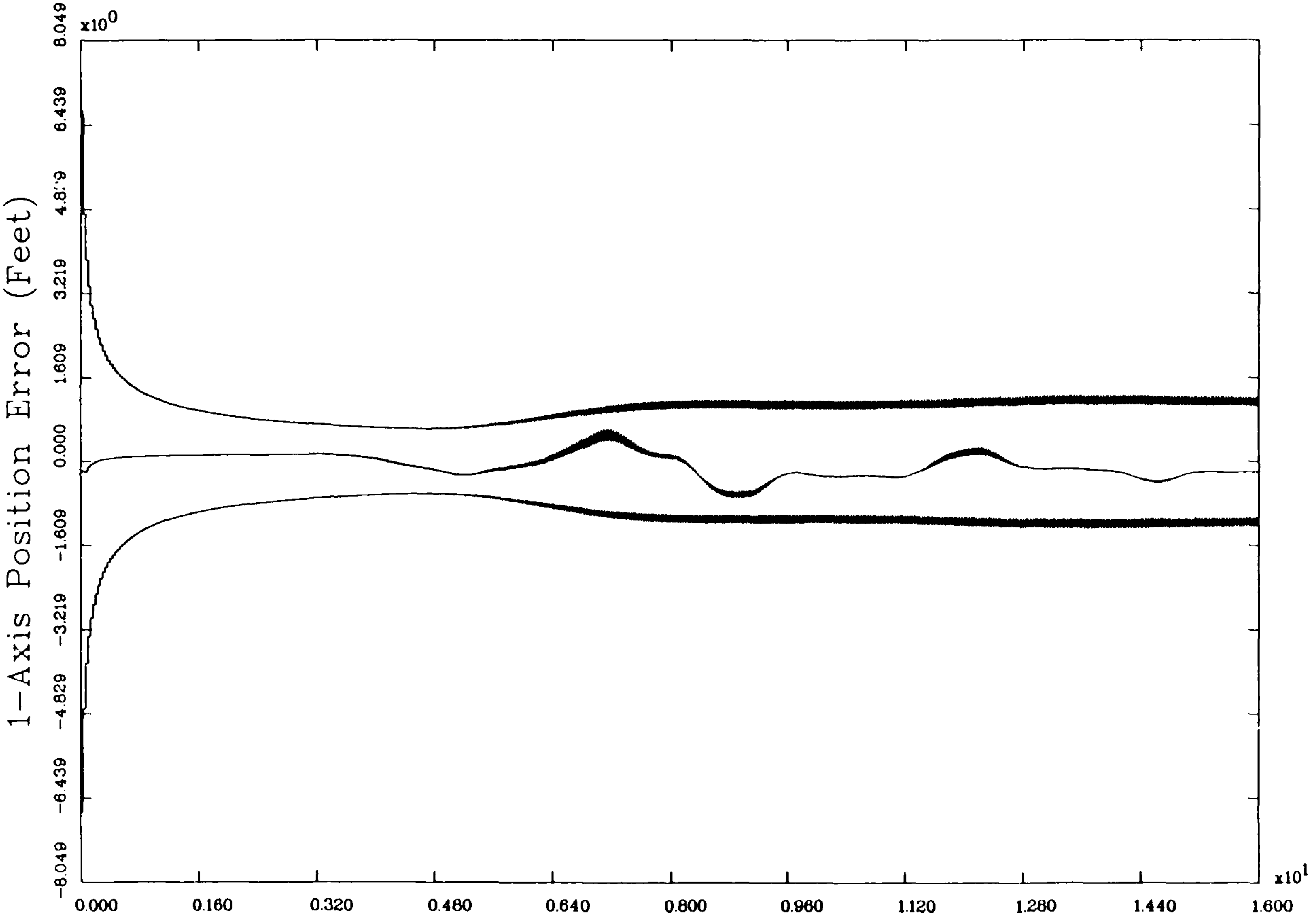


Figure 9.a. GM Filter, Trajectory 1, x_1

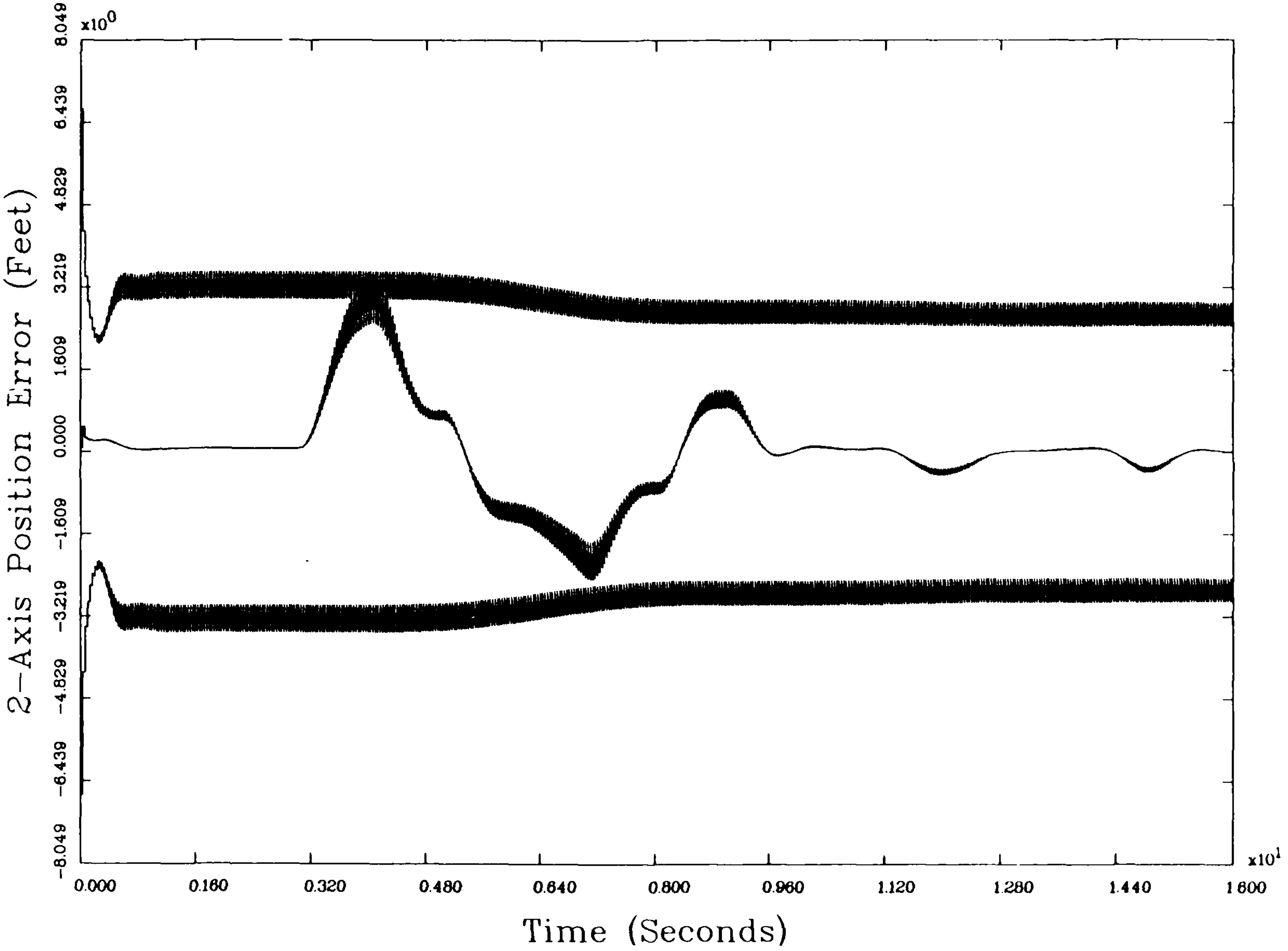


Figure 9.b. GM Filter, Trajectory 1, x_2

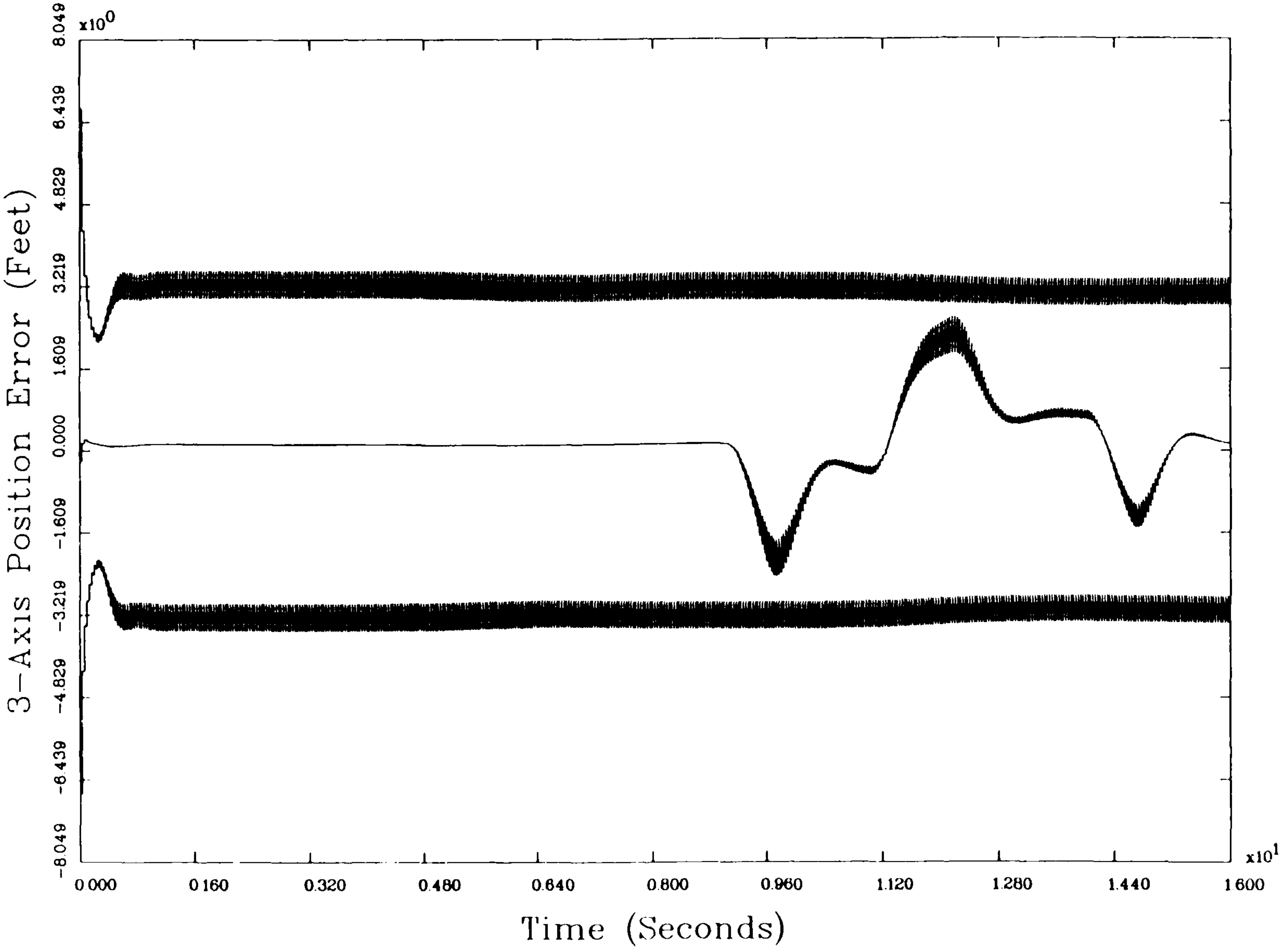


Figure 9.c. GM Filter, Trajectory 1, x_3

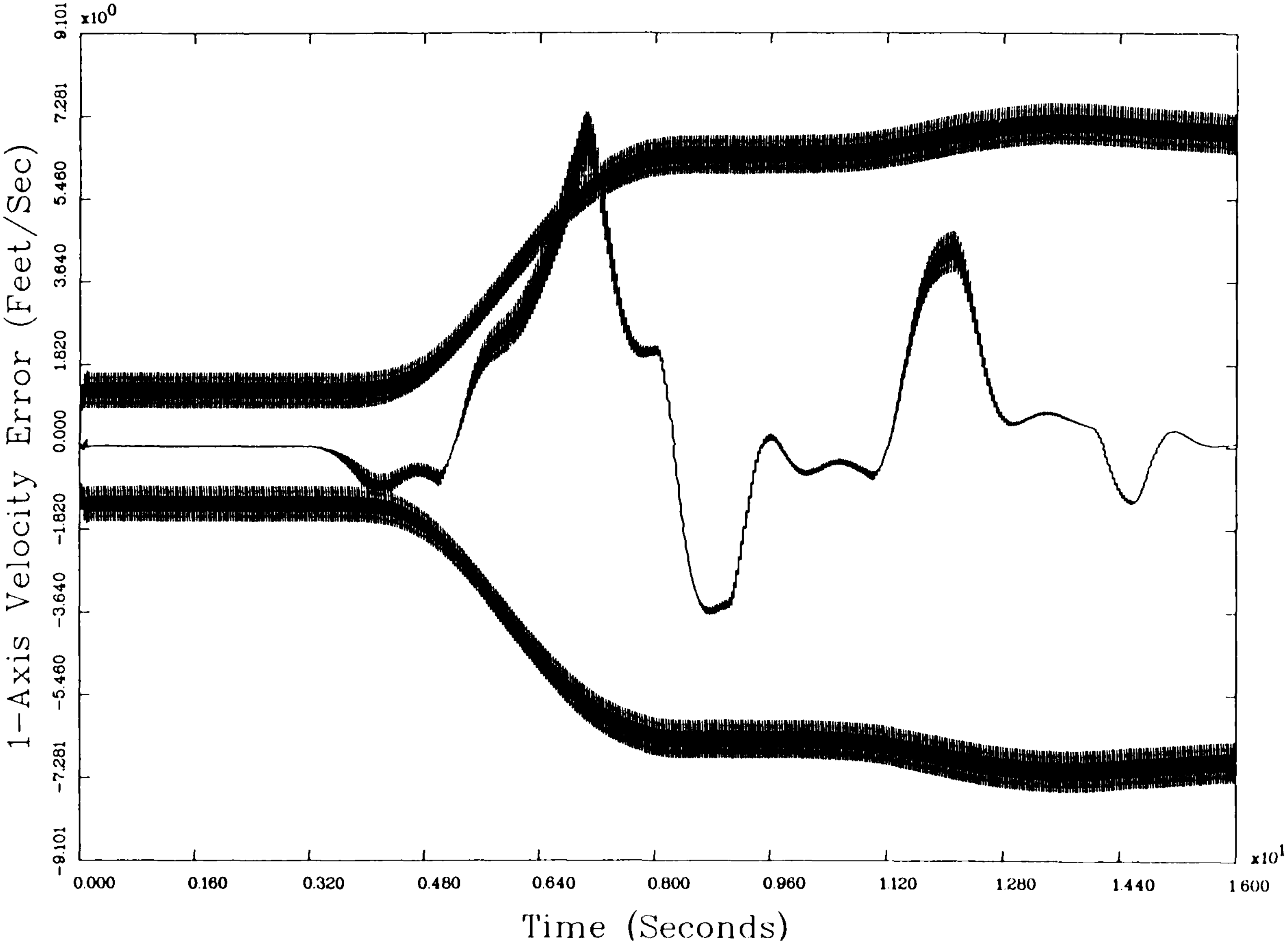


Figure 9.d. GM Filter, Trajectory 1, x_4

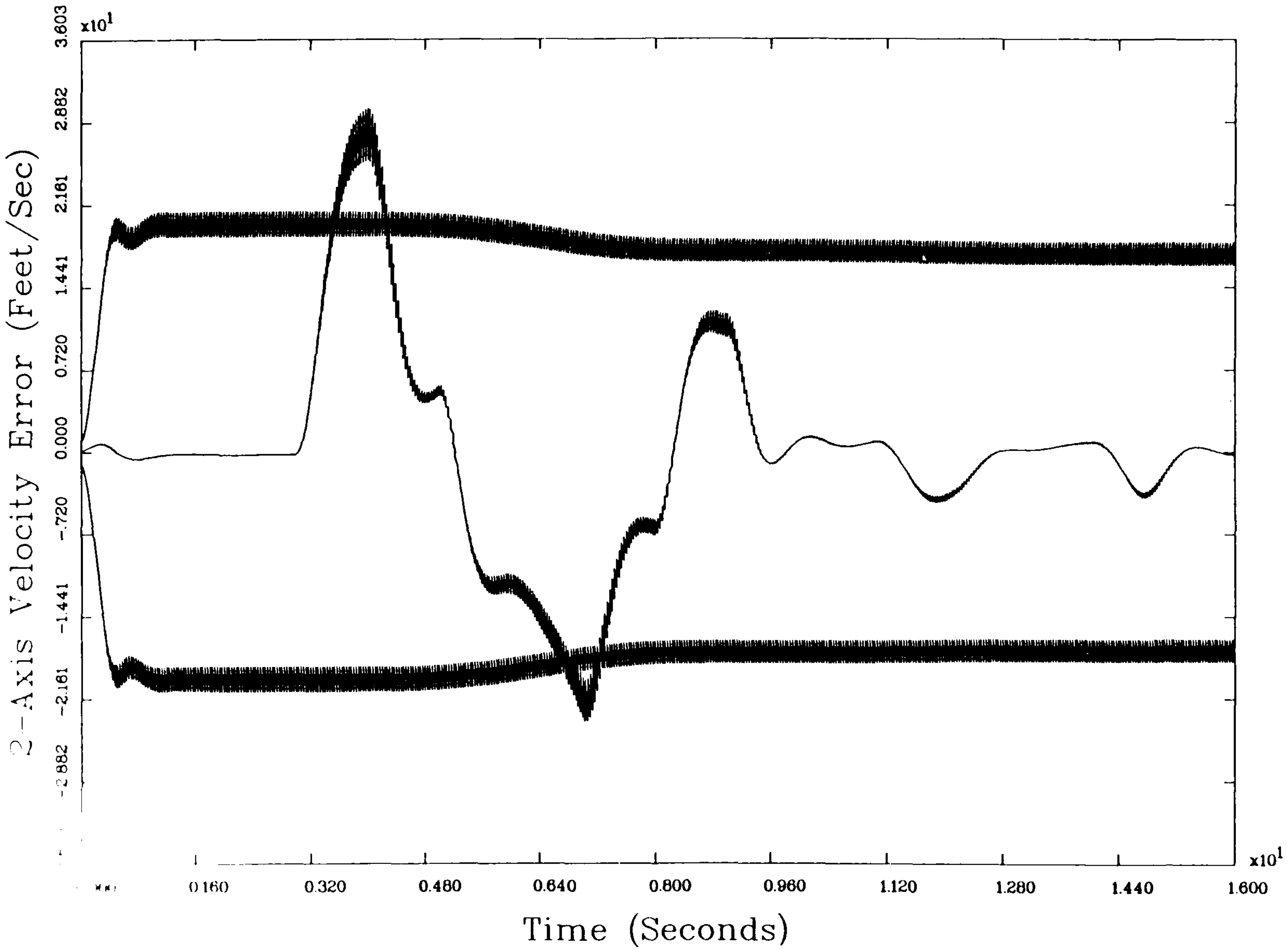
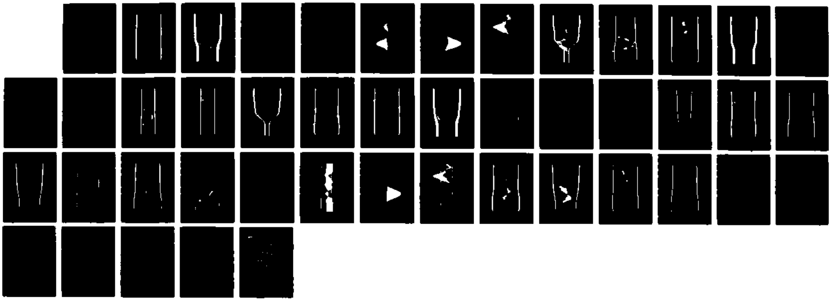
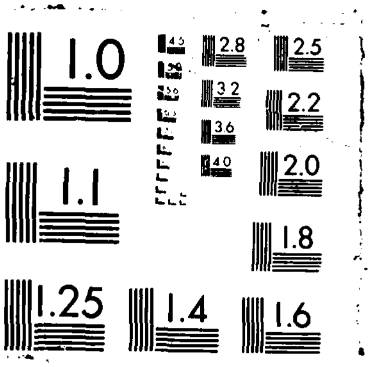


Figure 9.e. GM Filter, Trajectory 1, x_5

AD-A189 542 IMPLEMENTATION OF A TARGET STATE ESTIMATOR FOR THE 2/2
AIR-TO-AIR ATTACK MODE..(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. D L MICHALK
UNCLASSIFIED DEC 87 AFIT/GE/ENG/87D-44 F/G 19/5 NL




1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

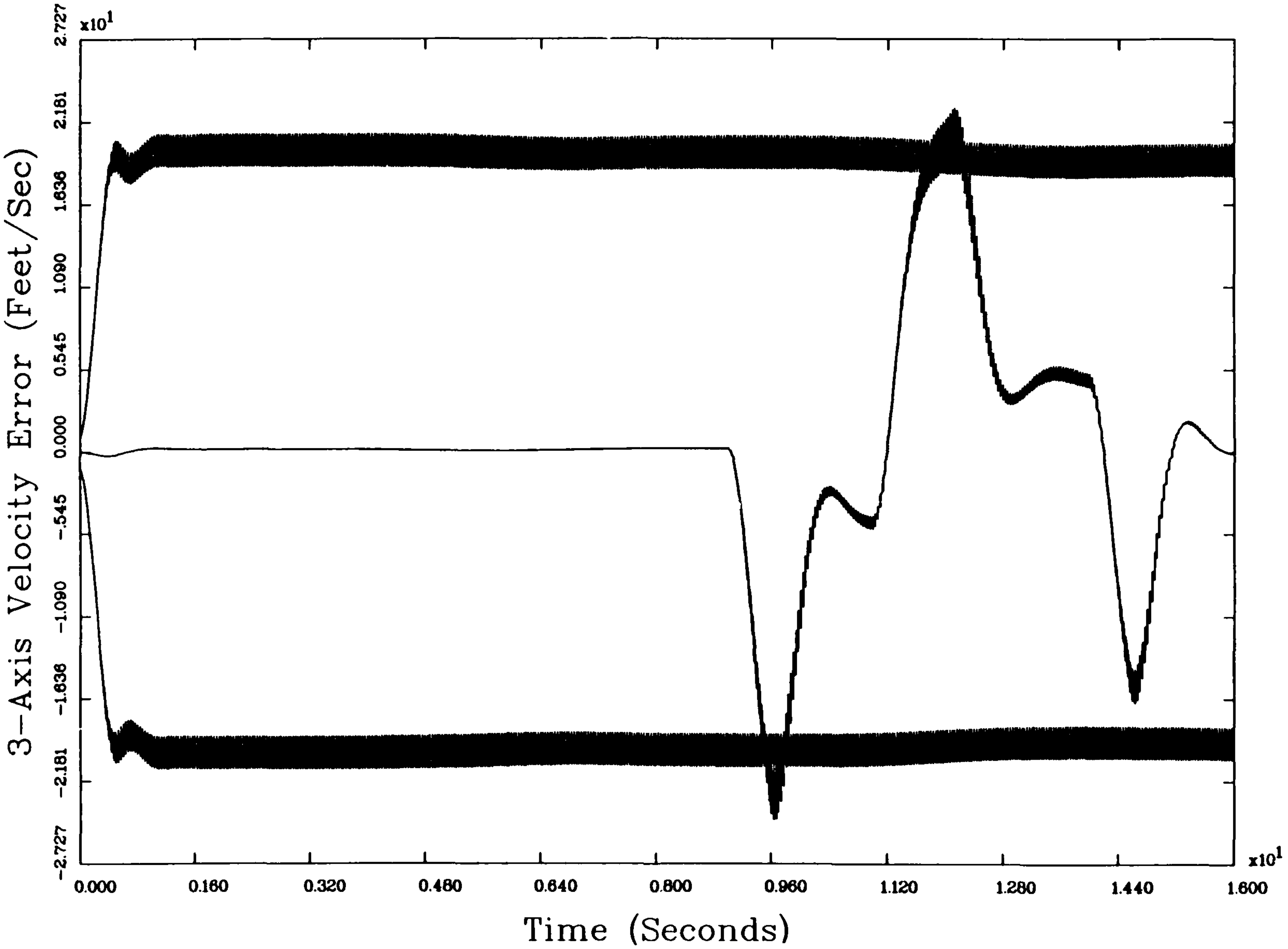


Figure 9.f. GM Filter, Trajectory 1, x_6

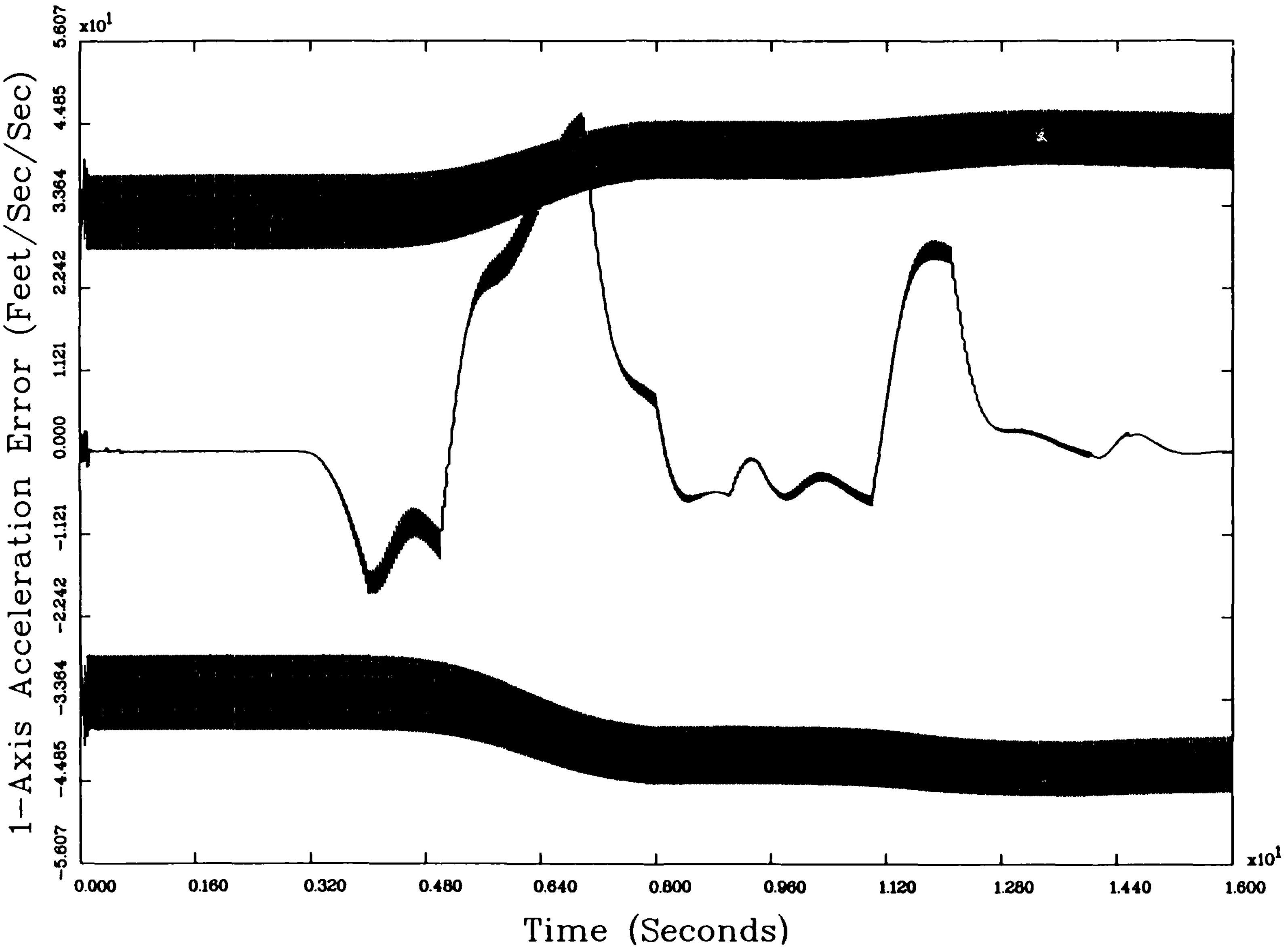


Figure 9.g. GM Filter, Trajectory 1, x_7

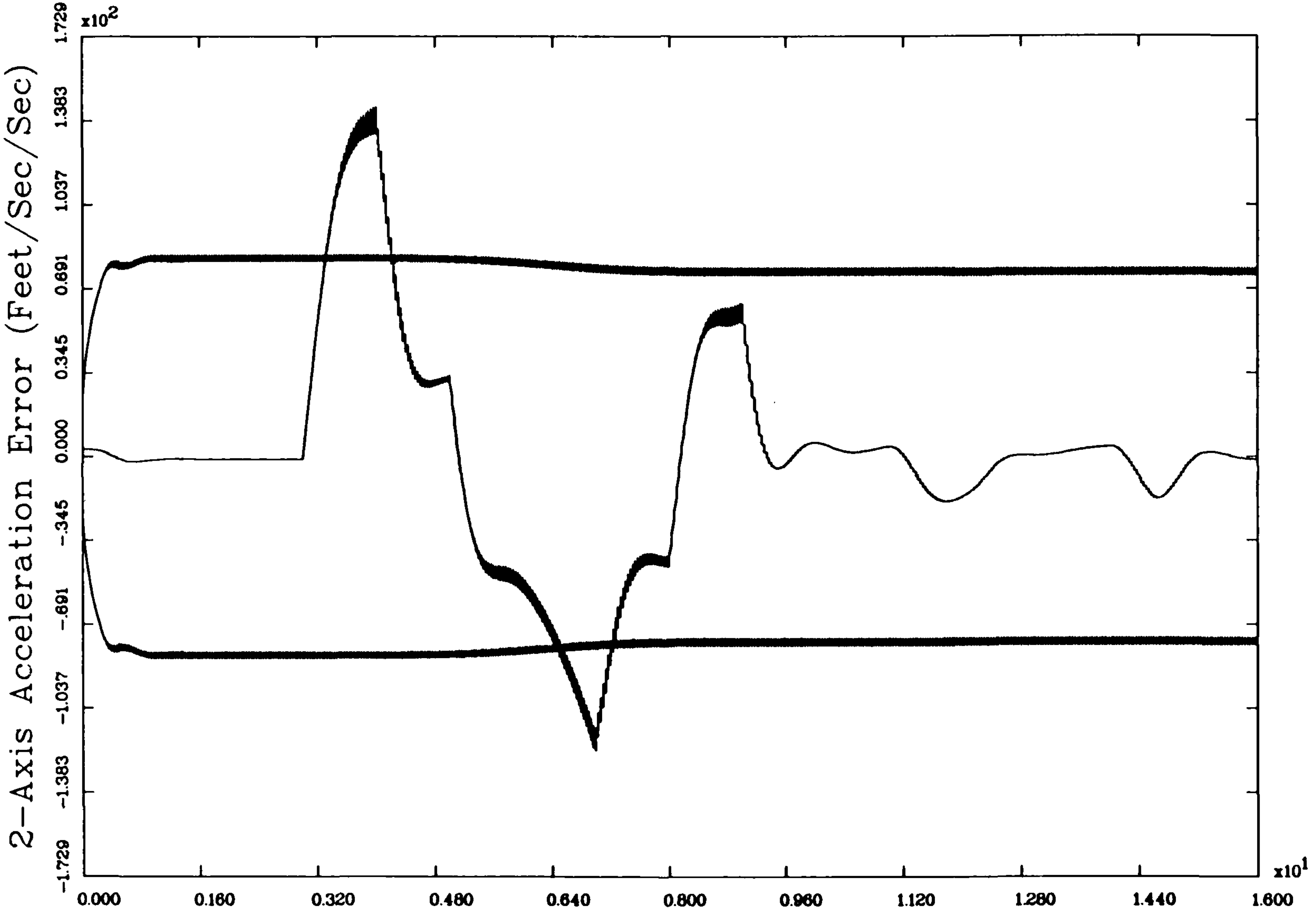


Figure 9.h. GM Filter, Trajectory 1, x_8

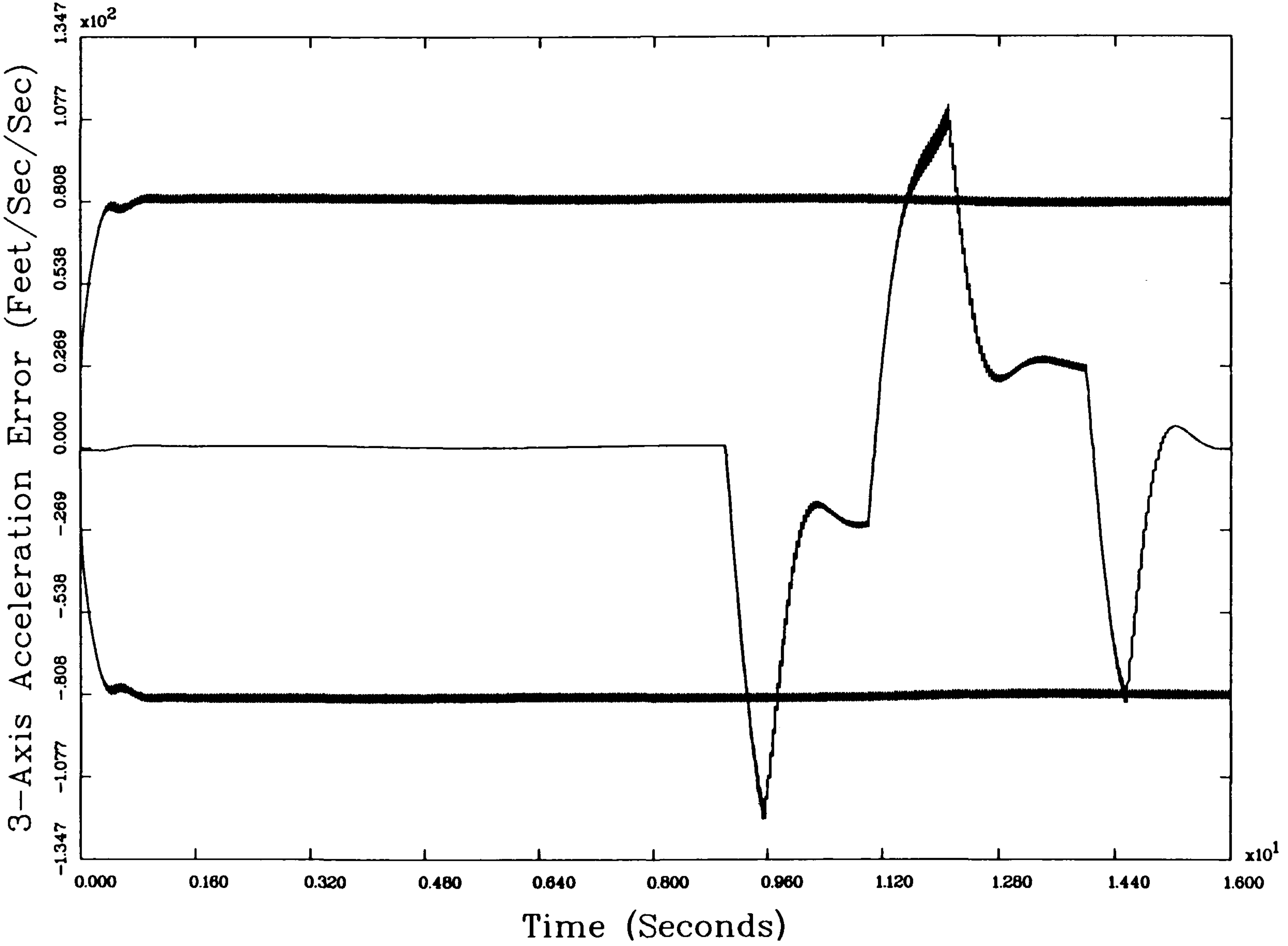


Figure 9.1. GM Filter, Trajectory 1, x_9

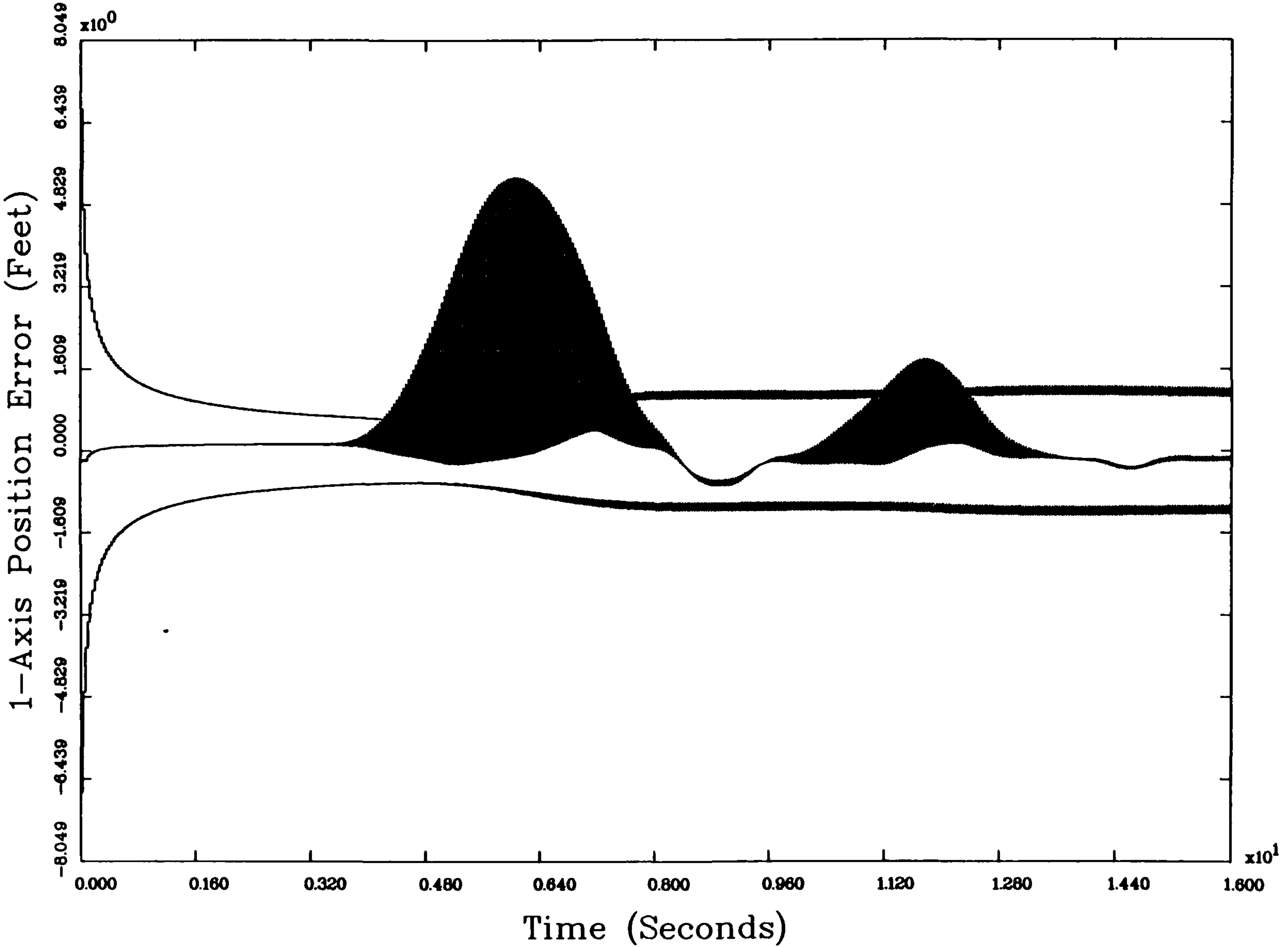


Figure 10.a. GM Filter, U-D Implementation, Trajectory 1, x_1

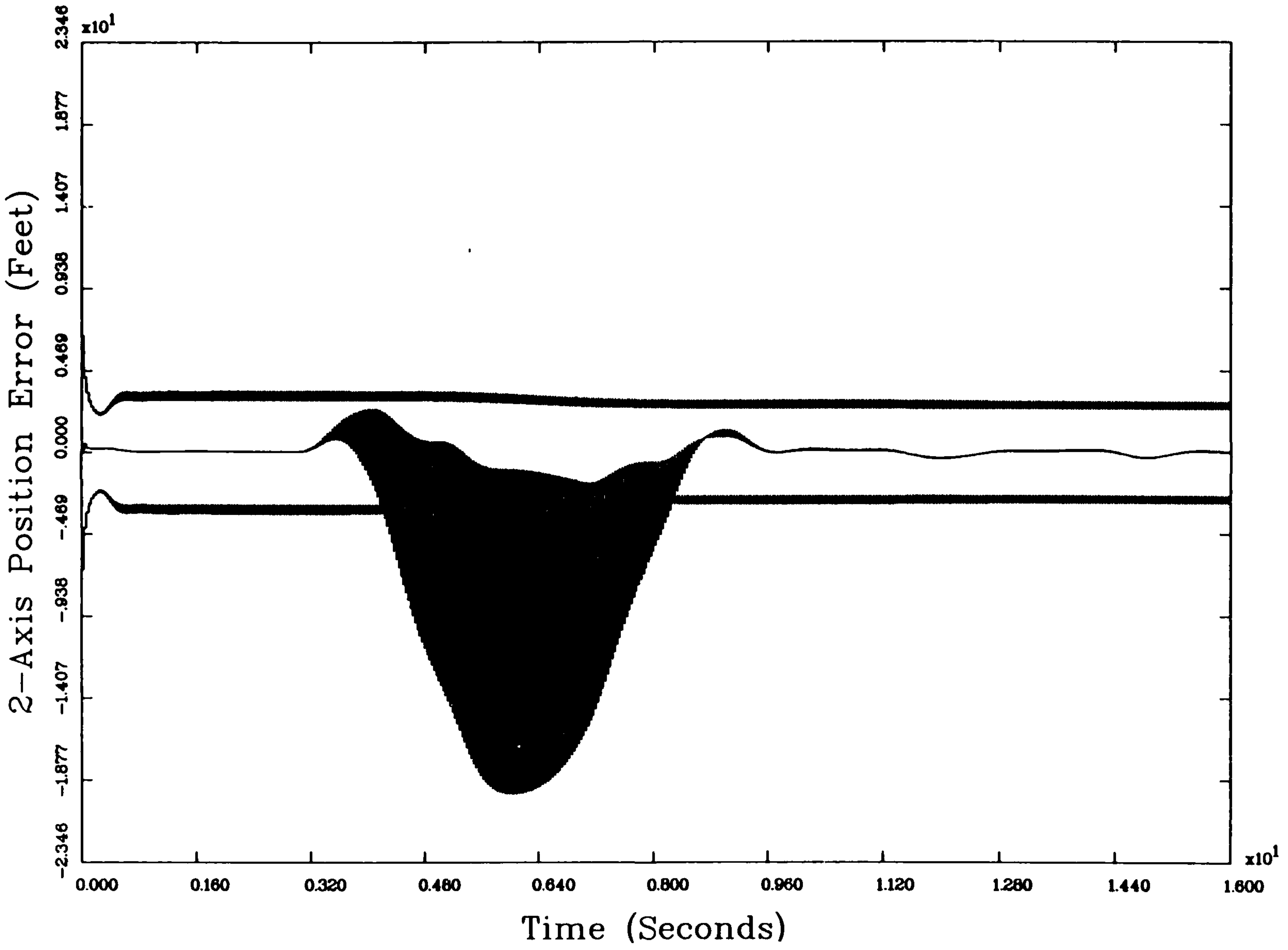


Figure 10.b. GM Filter, U-D Implementation, Trajectory 1, x_2

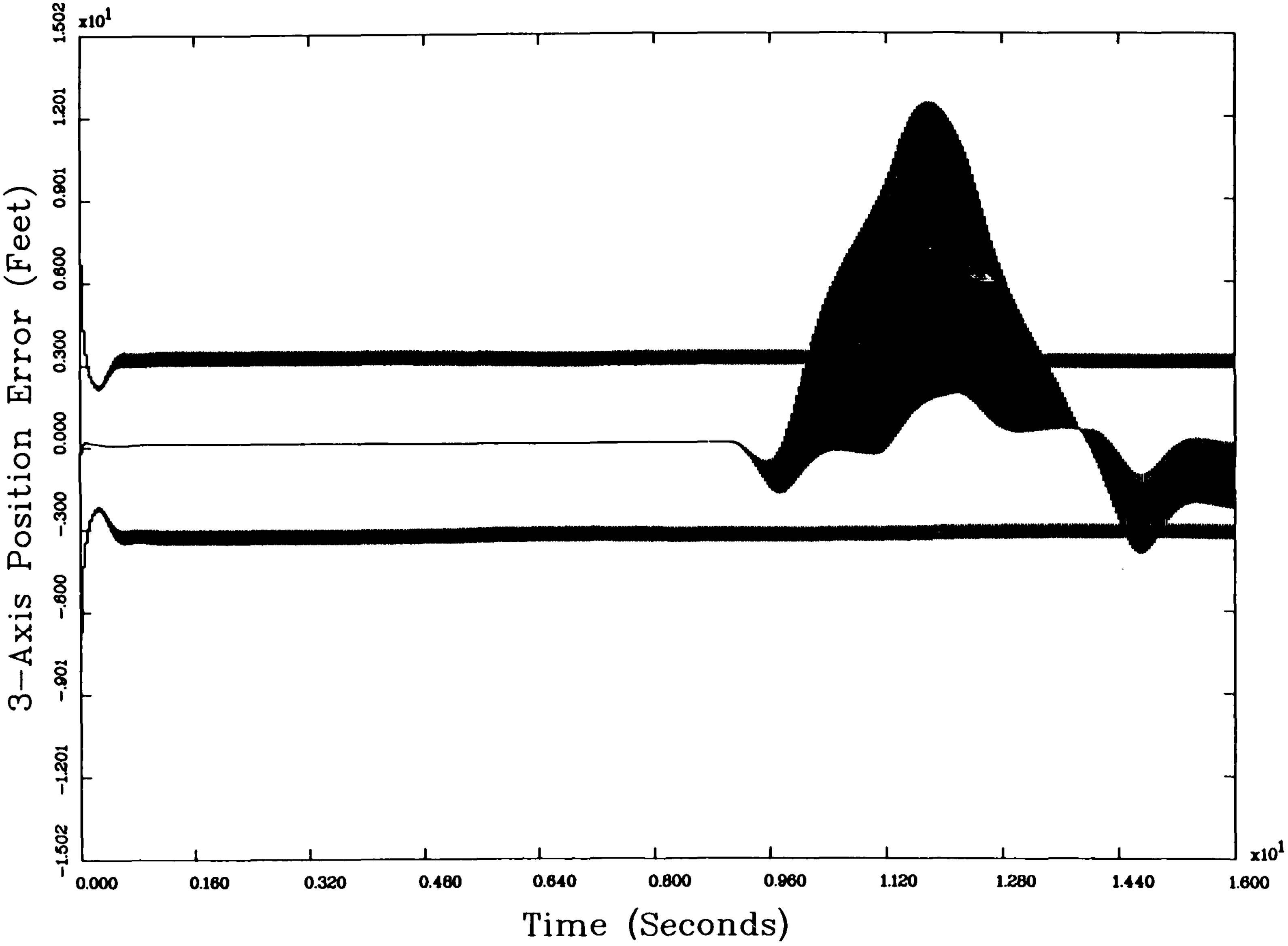


Figure 10.c. GM Filter, U-D Implementation, Trajectory 1, x_3

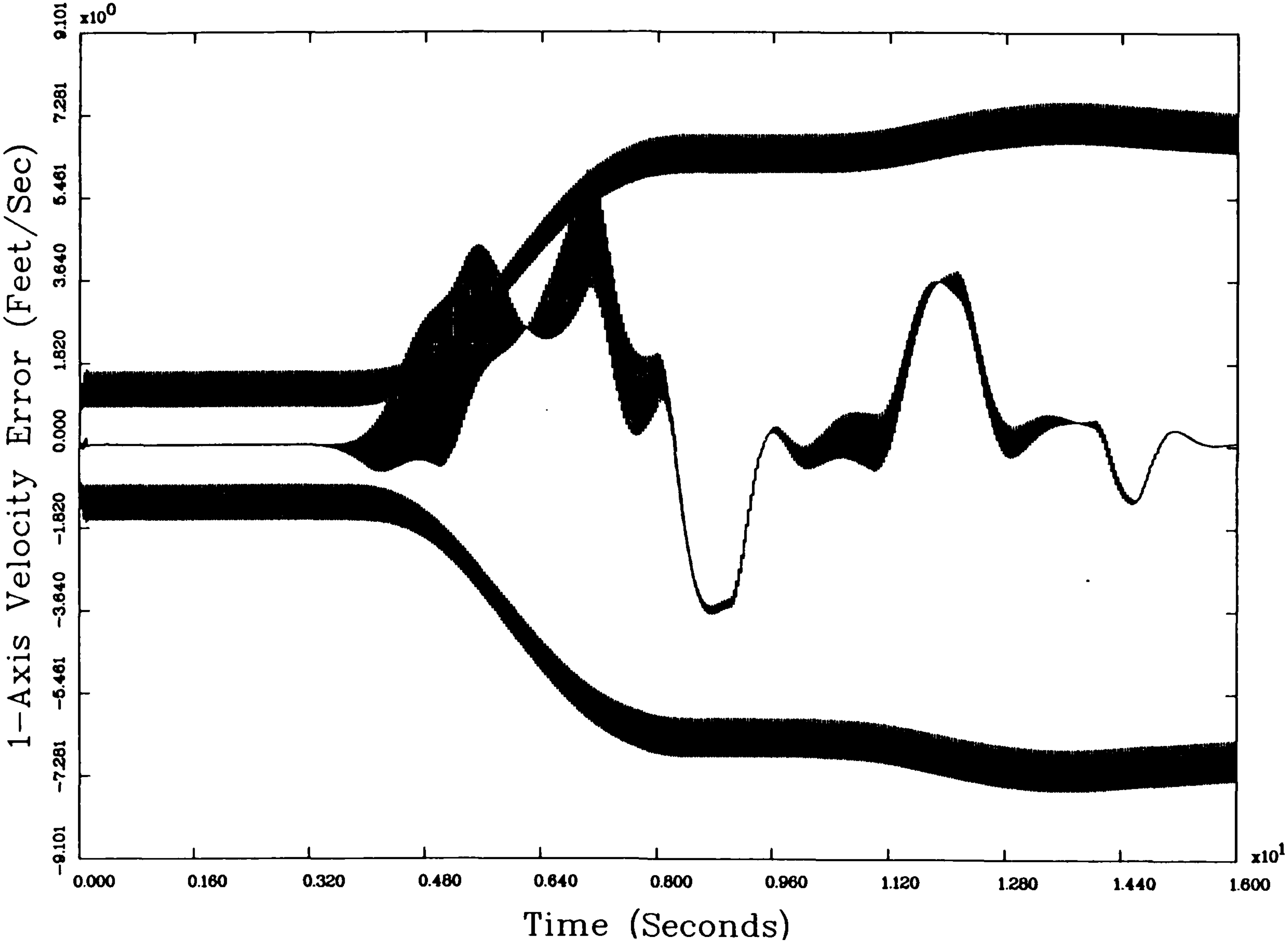


Figure 10.d. GM Filter, U-D Implementation, Trajectory 1, x_4

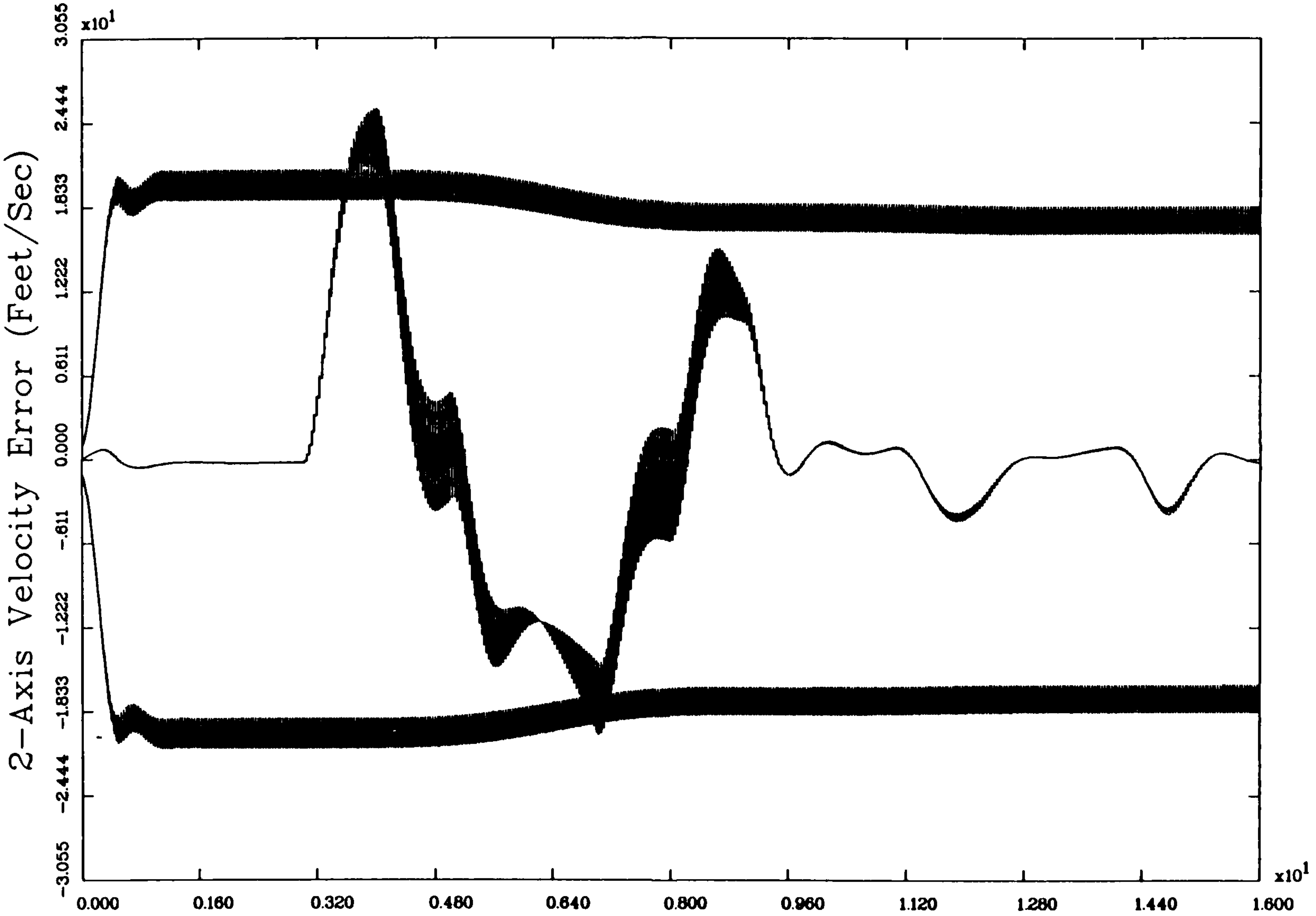


Figure 10.e. GM Filter, U-D Implementation, Trajectory 1, x_5

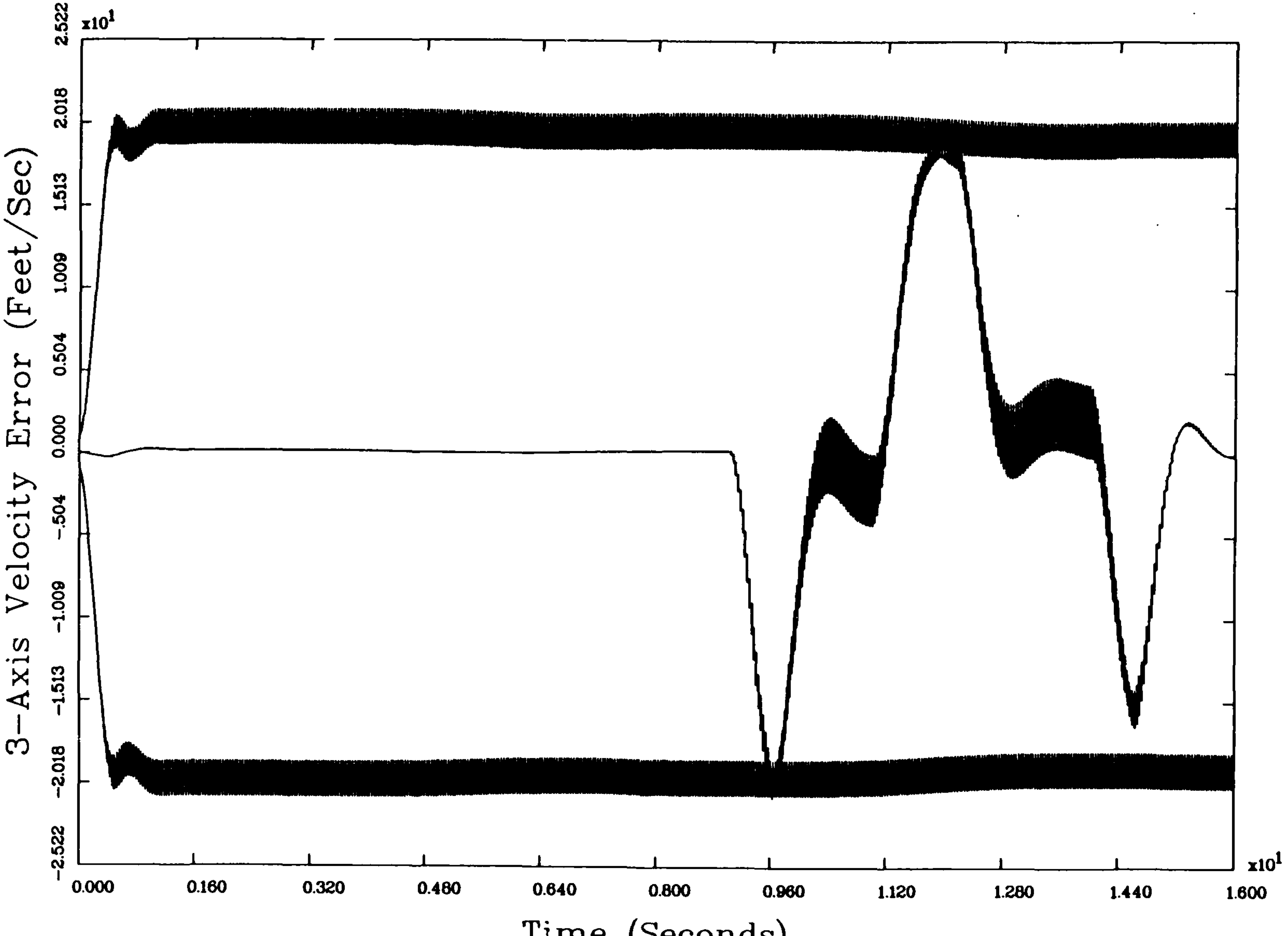


Figure 10.f. GM Filter, U-D Implementation, Trajectory 1, x_6

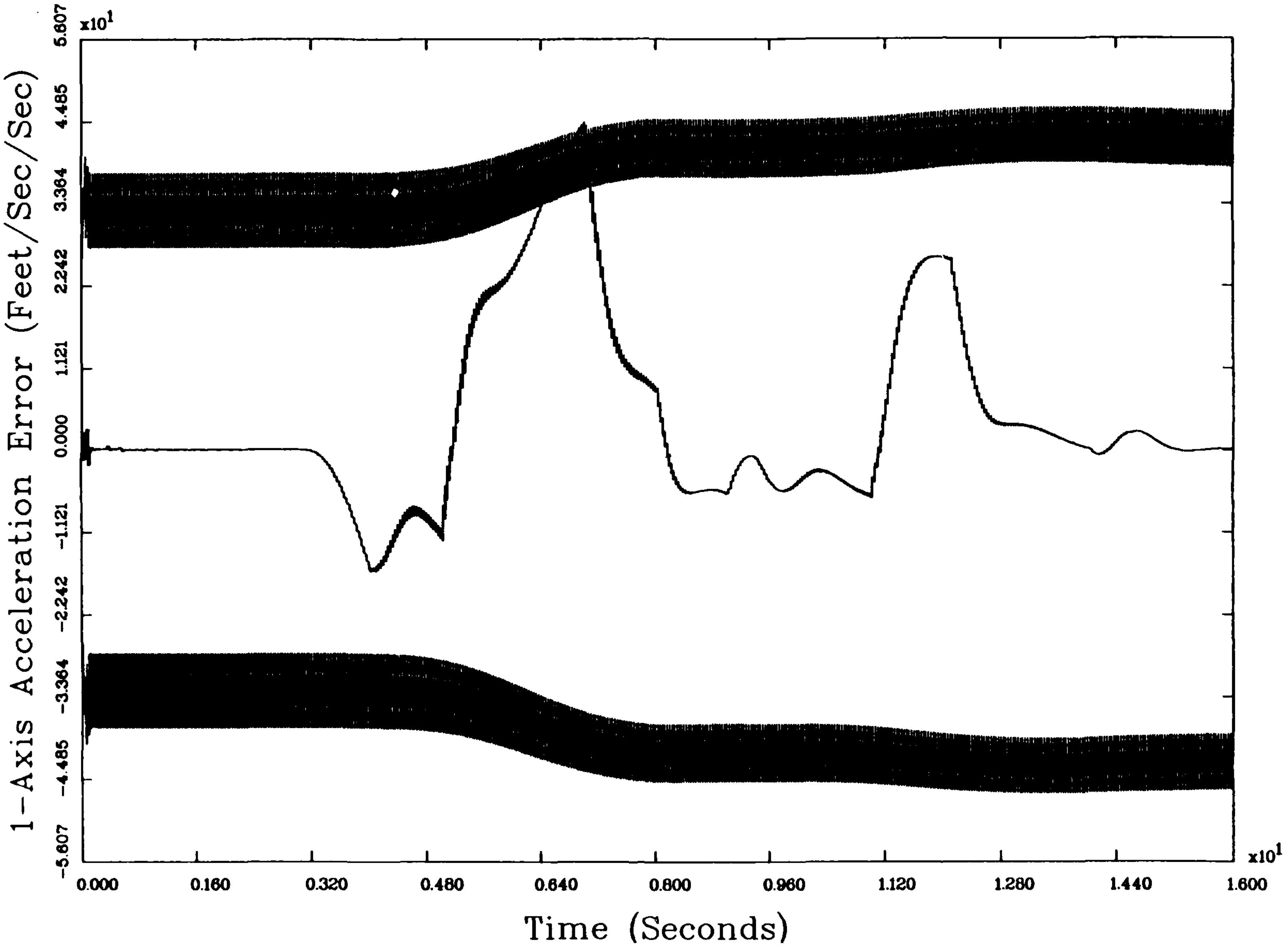


Figure 10.g. GM Filter, U-D Implementation, Trajectory 1, x_7

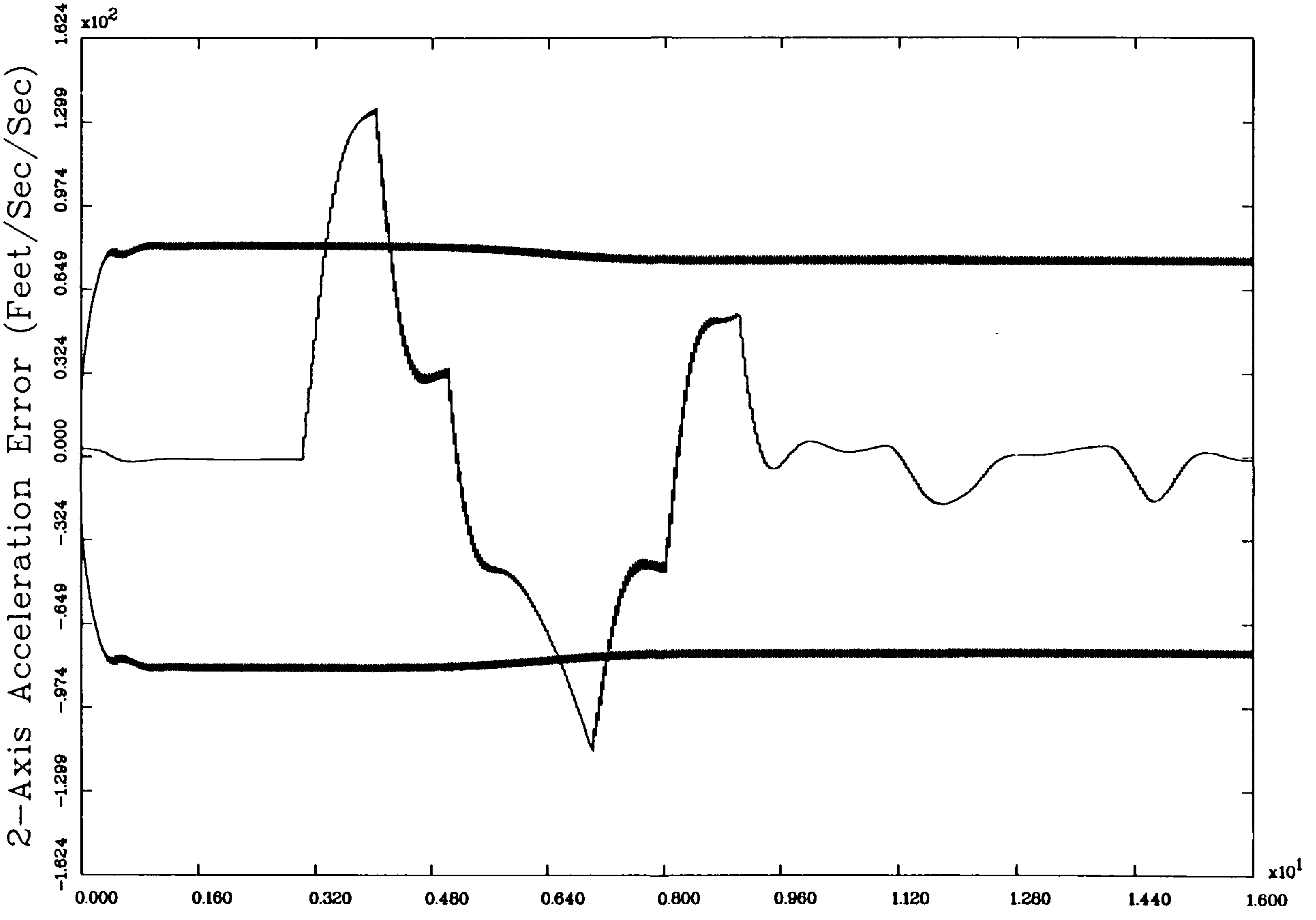


Figure 10.h. GM Filter, U-D Implementation, Trajectory 1, x_8

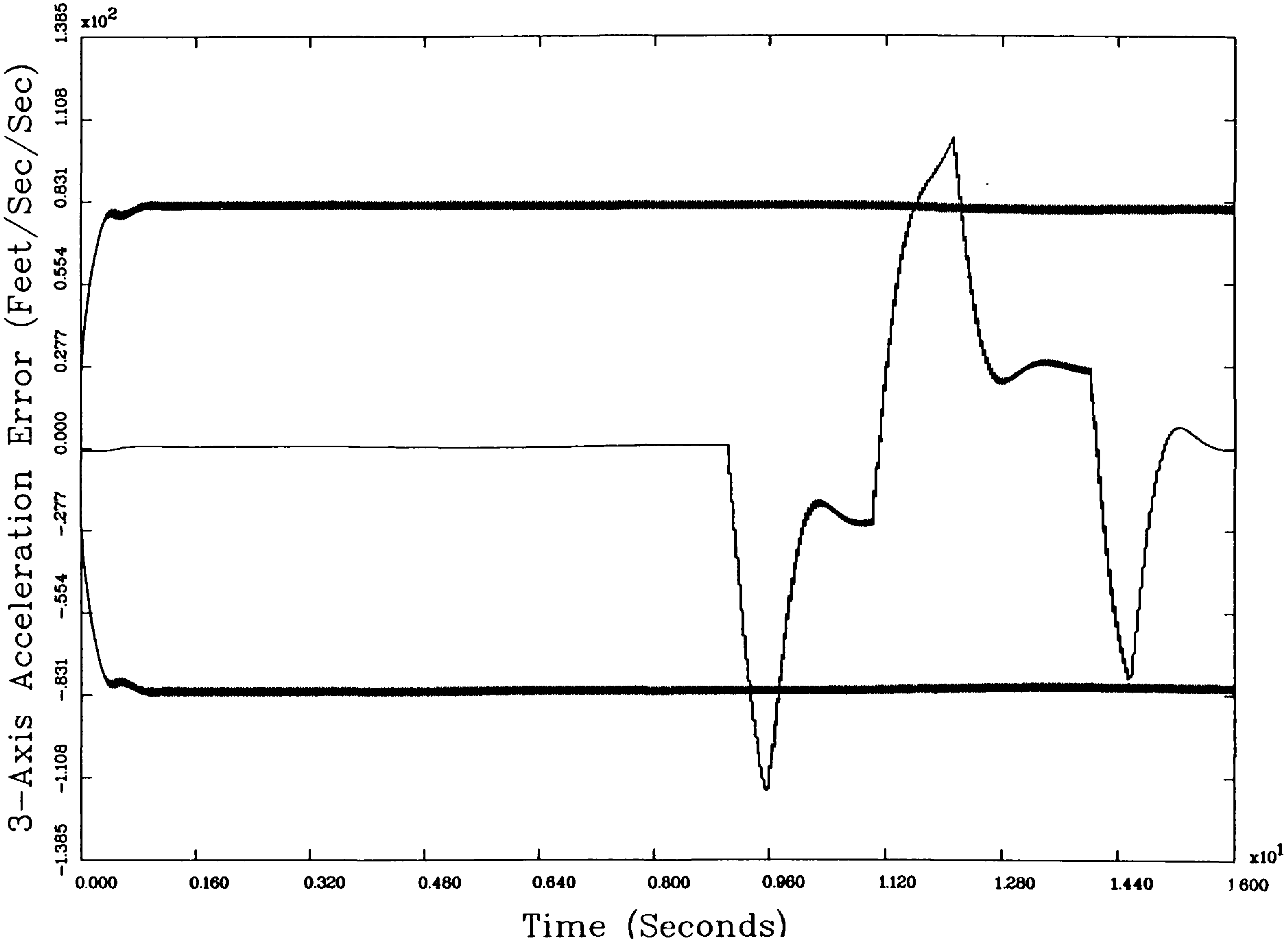


Figure 10.1. GM Filter, U-D Implementation, Trajectory 1, x_9

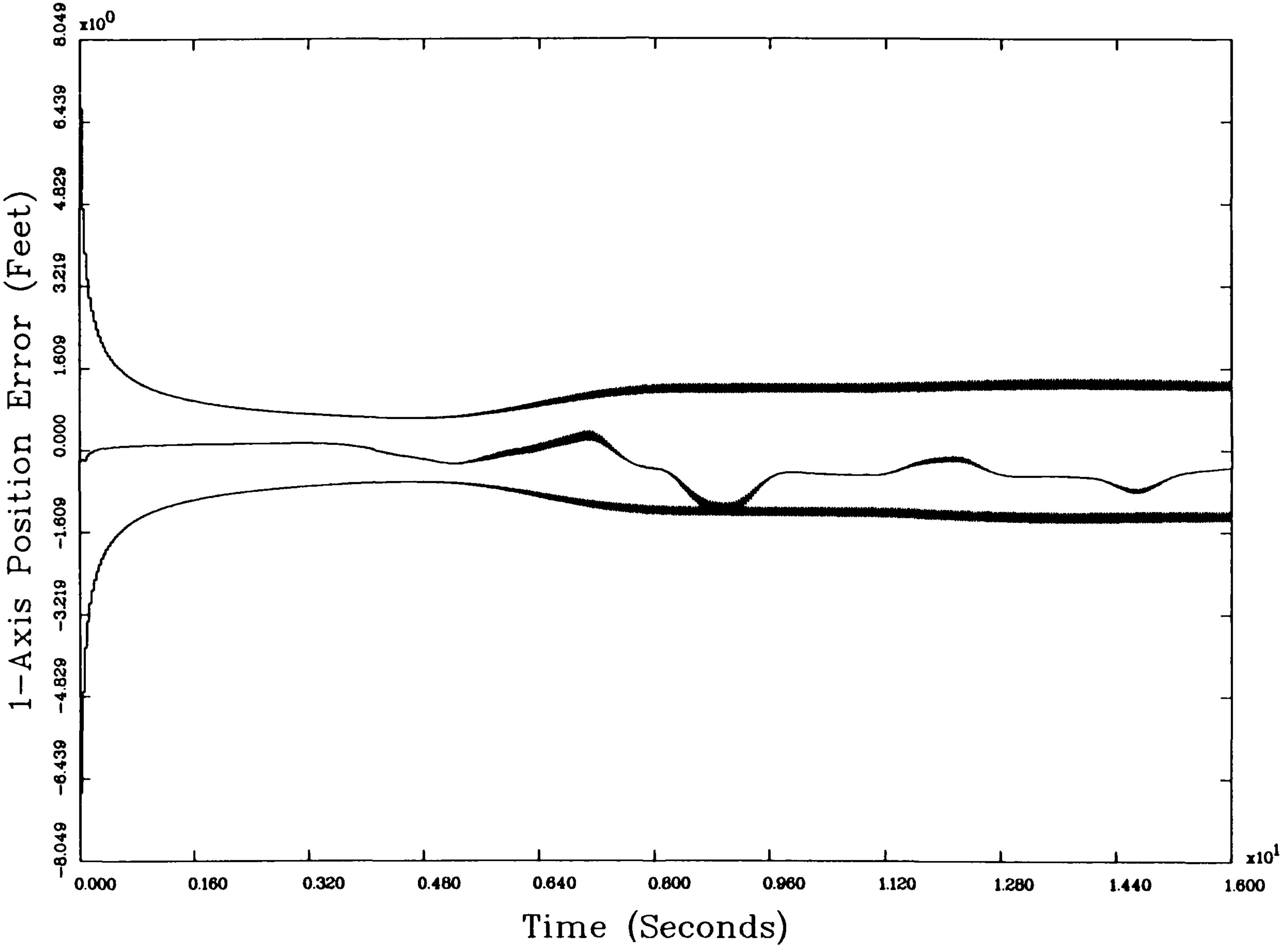


Figure 11.a. CTR Filter, Trajectory 2, x_1

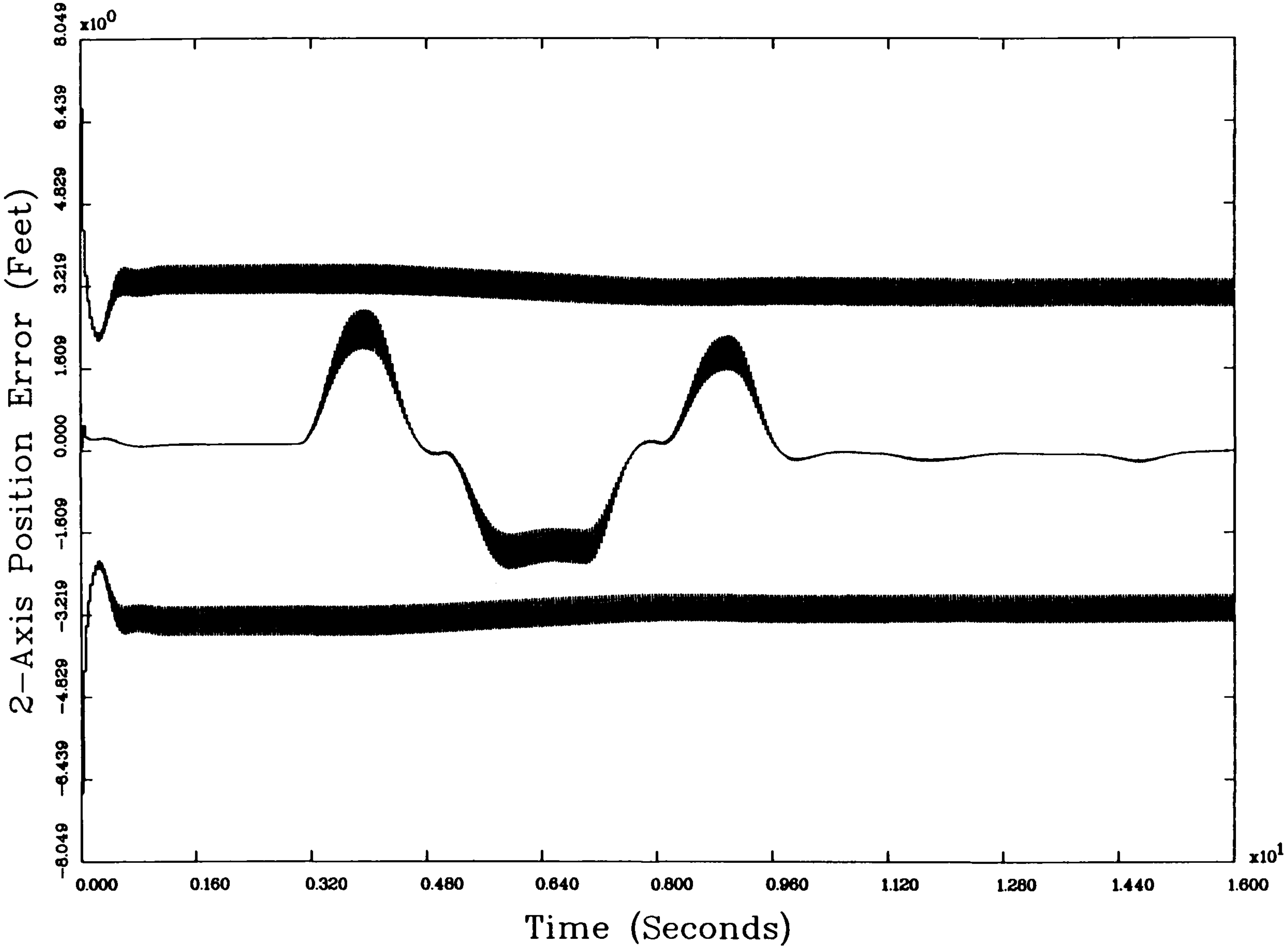


Figure 11.b. CTR Filter, Trajectory 2, x_2

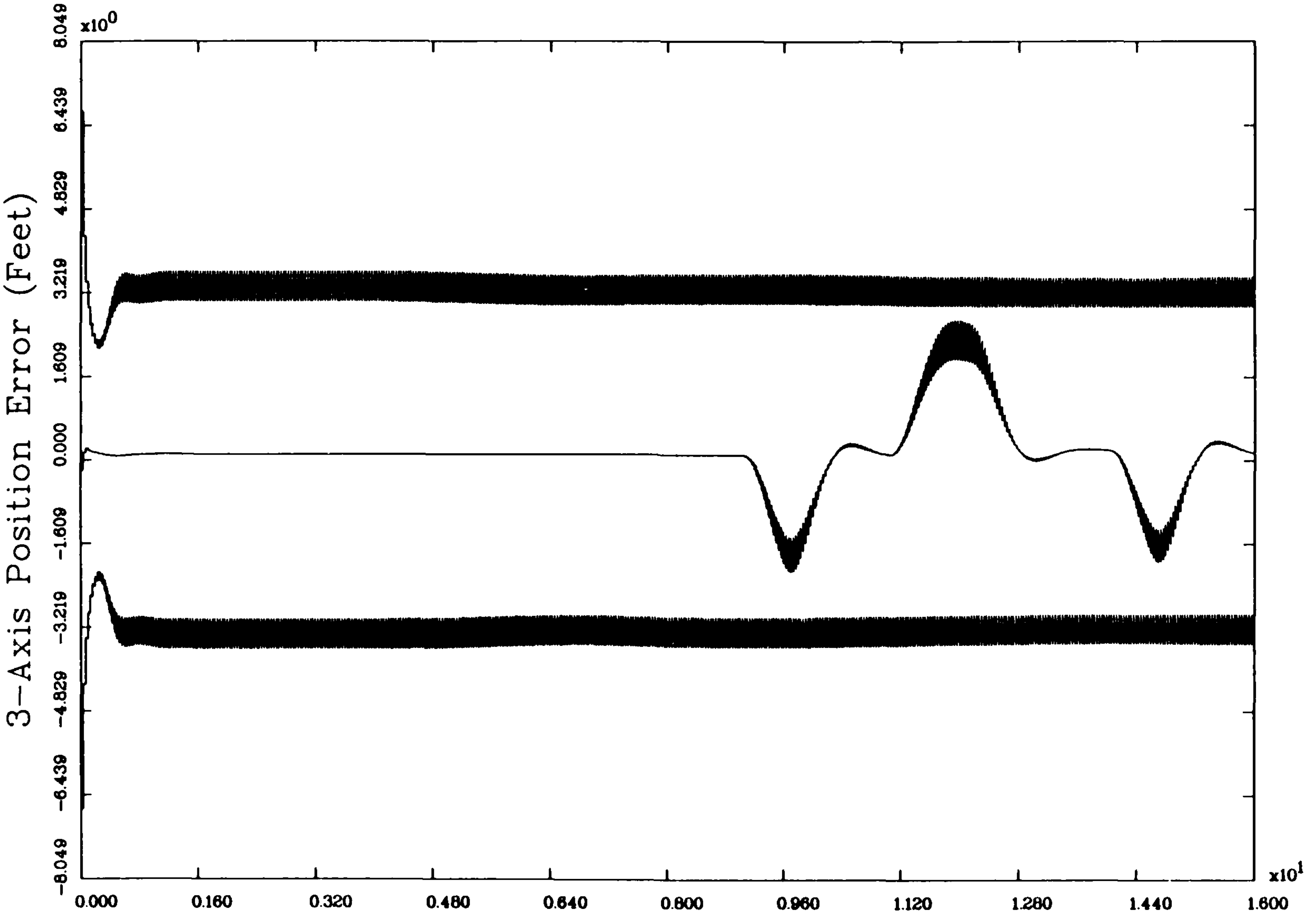


Figure 11.c. CTR Filter, Trajectory 2, x_3

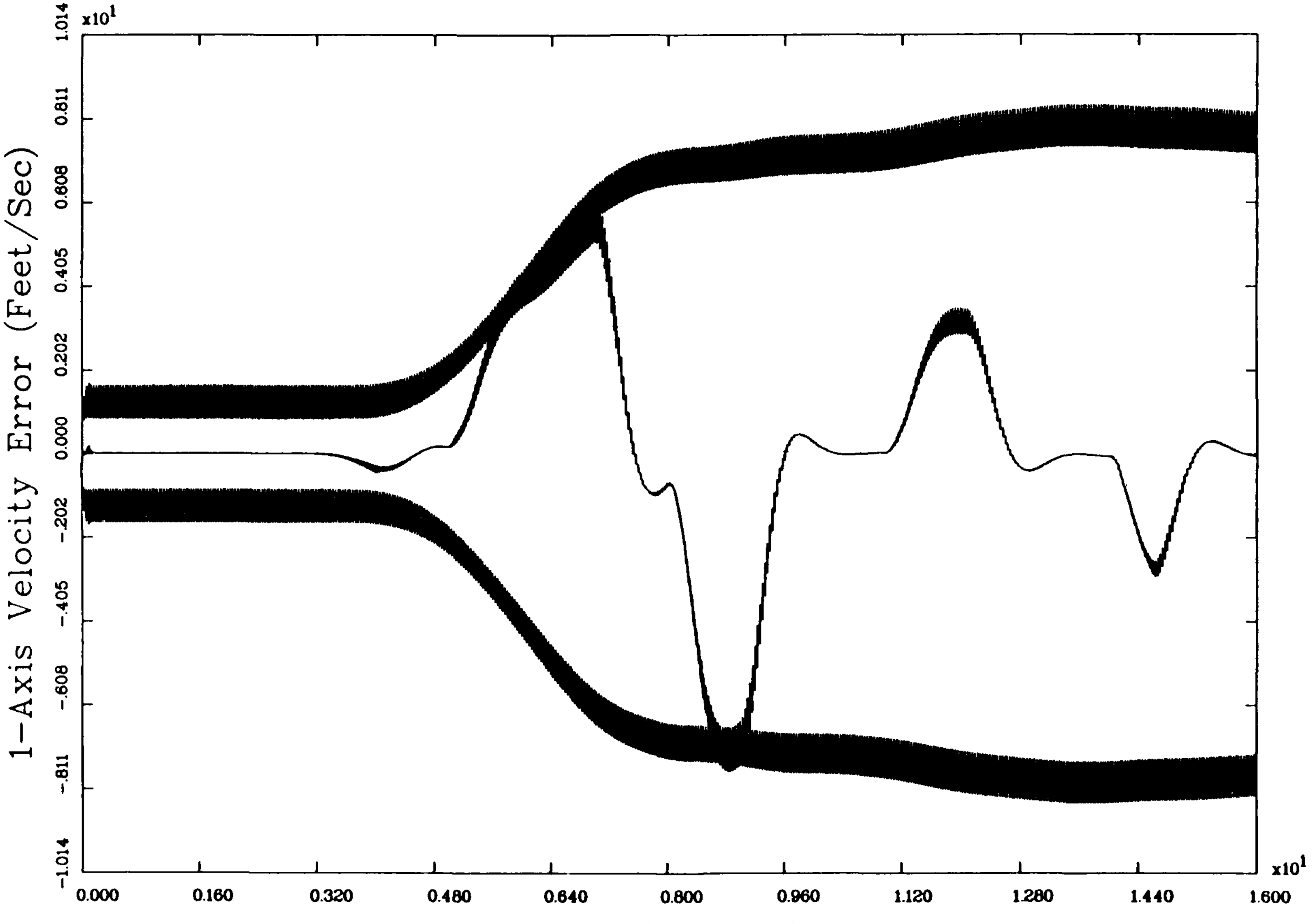


Figure 11.d. CTR Filter, Trajectory 2, x_4

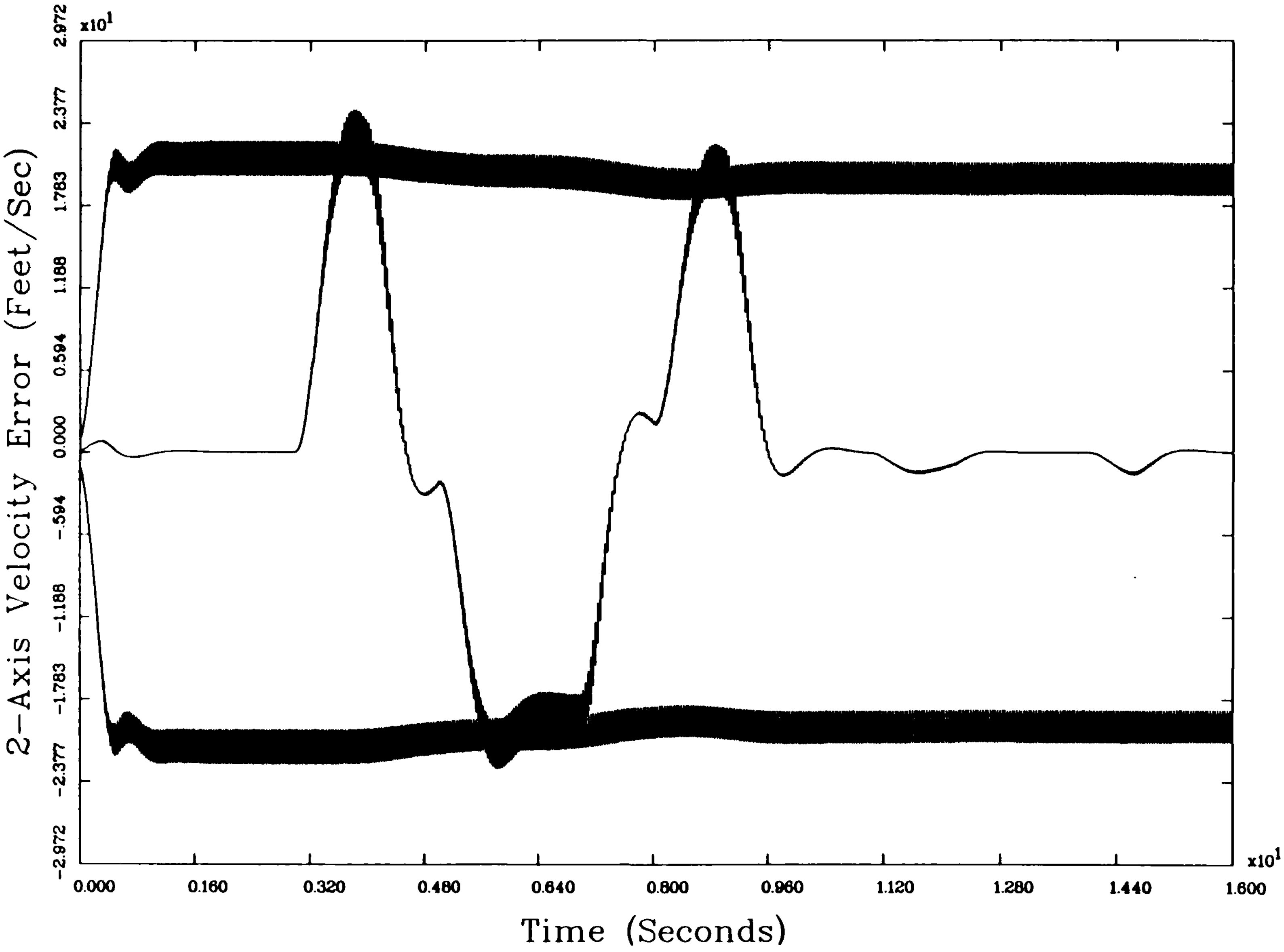


Figure 11.e. CTR Filter, Trajectory 2, x_5

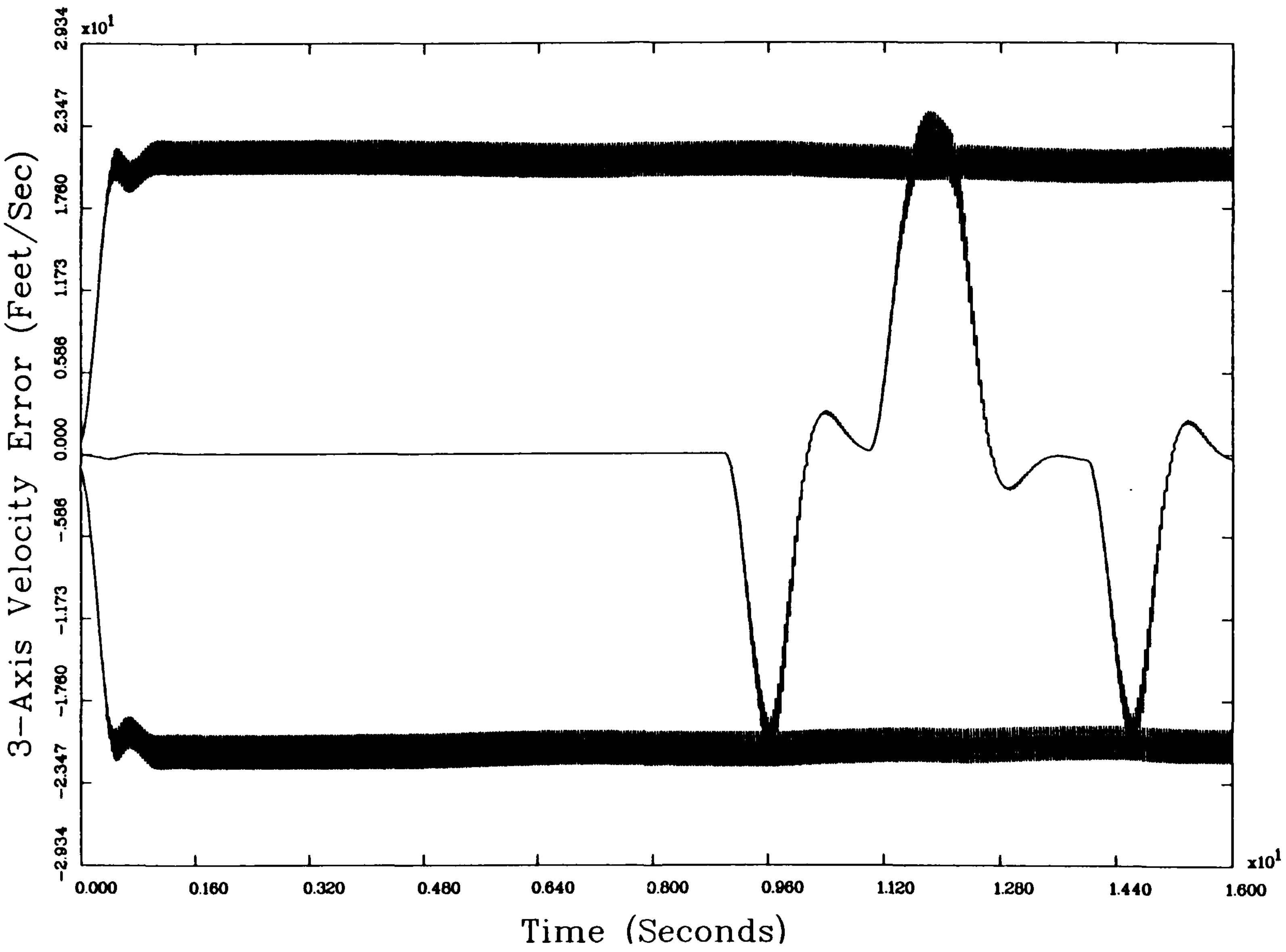


Figure 11.f. CTR Filter, Trajectory 2, x_6

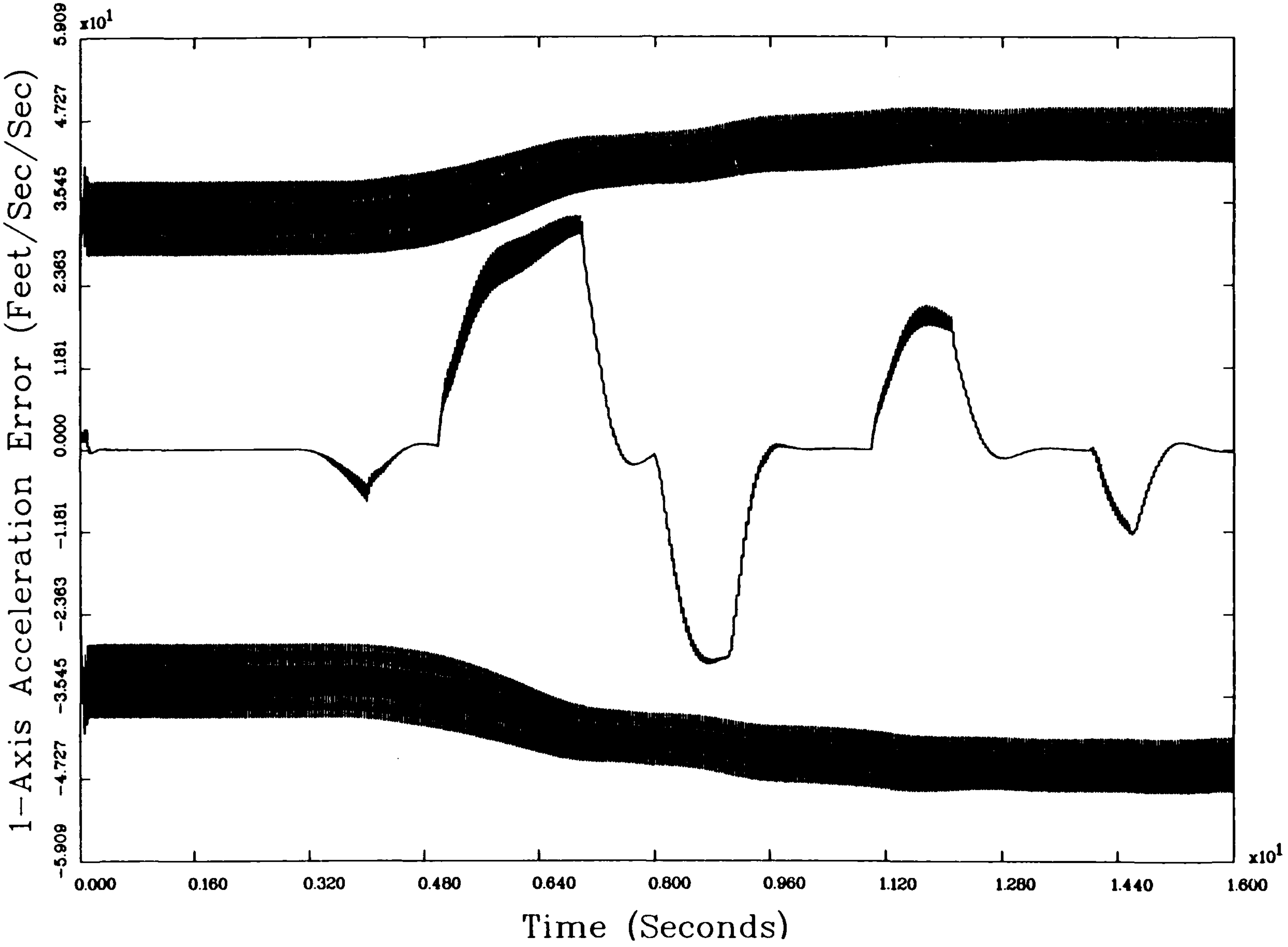


Figure 11.g. CTR Filter, Trajectory 2, x_7

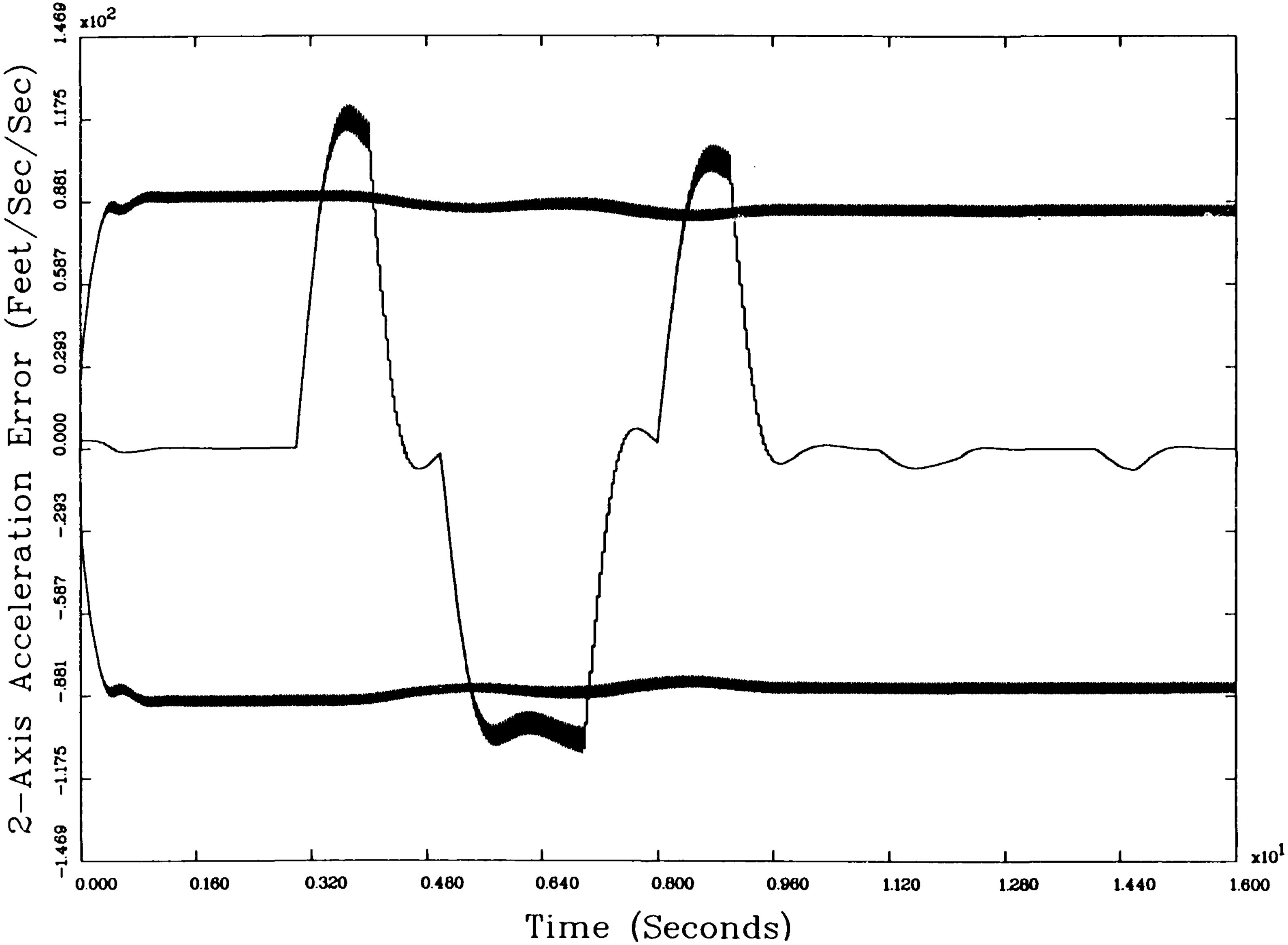


Figure 11.h. CTR Filter, Trajectory 2, x_8

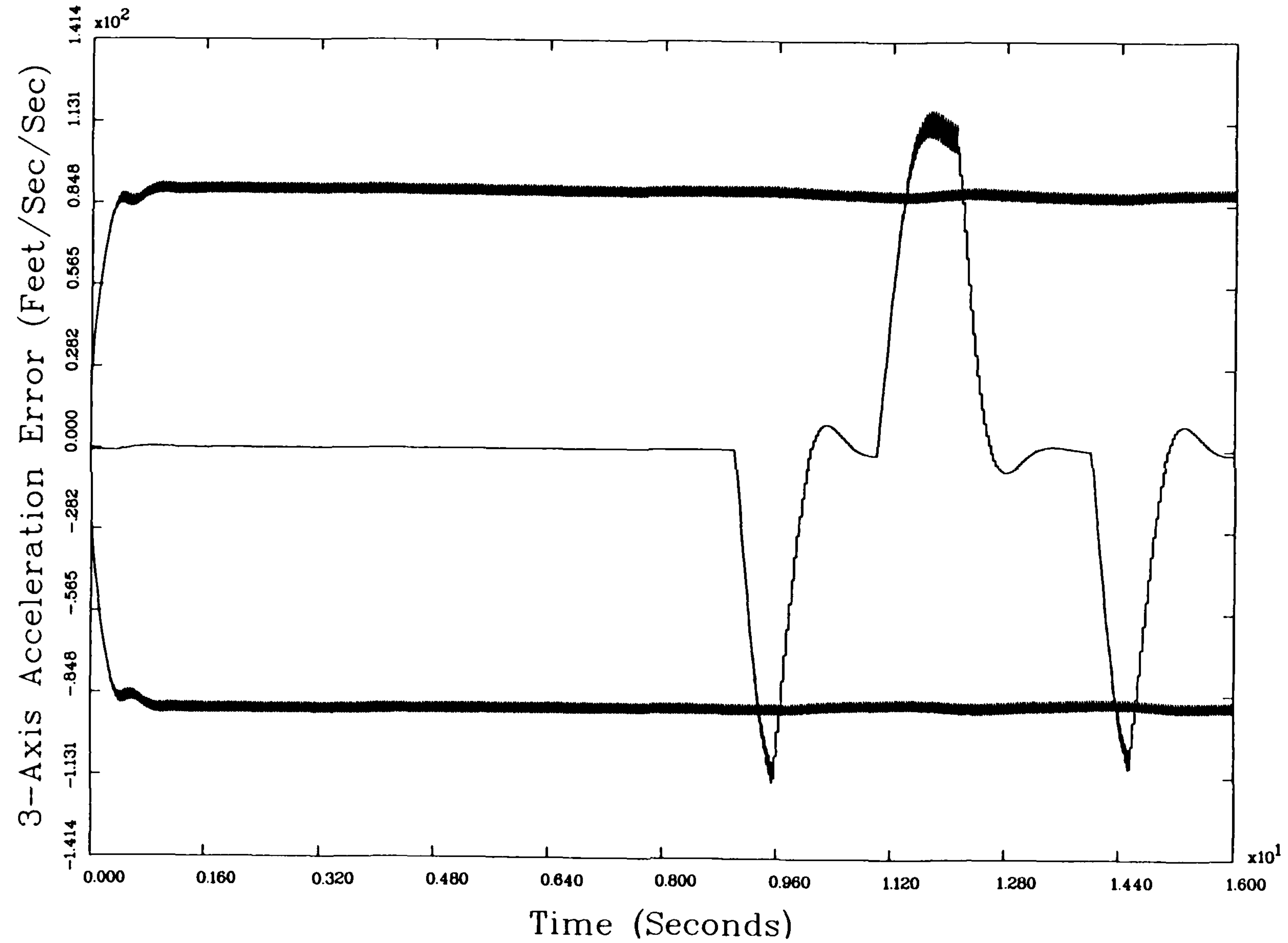


Figure 11.1. CTR Filter, Trajectory 2, x_9

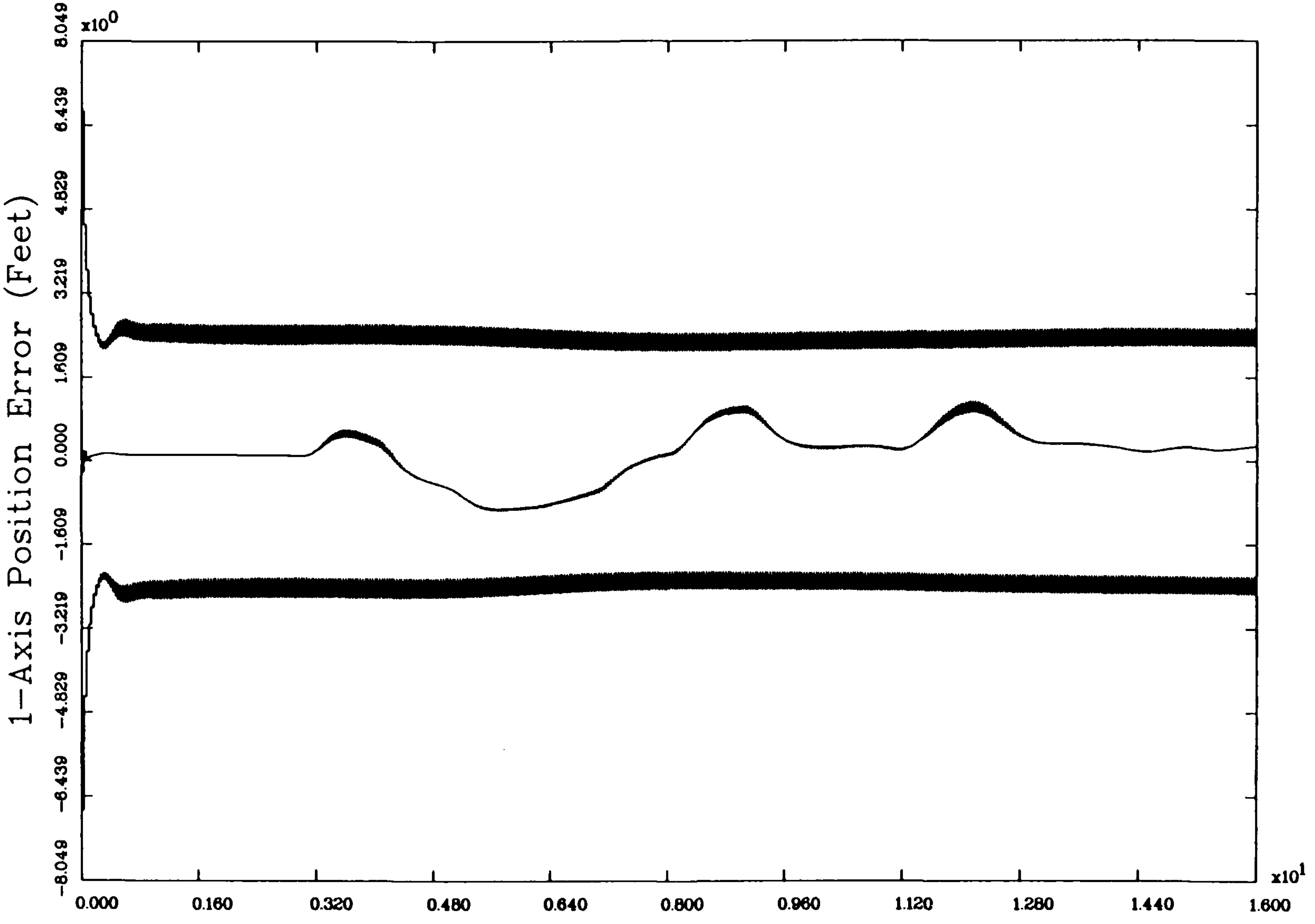


Figure 12.a. GM Filter, Trajectory 2, x_1

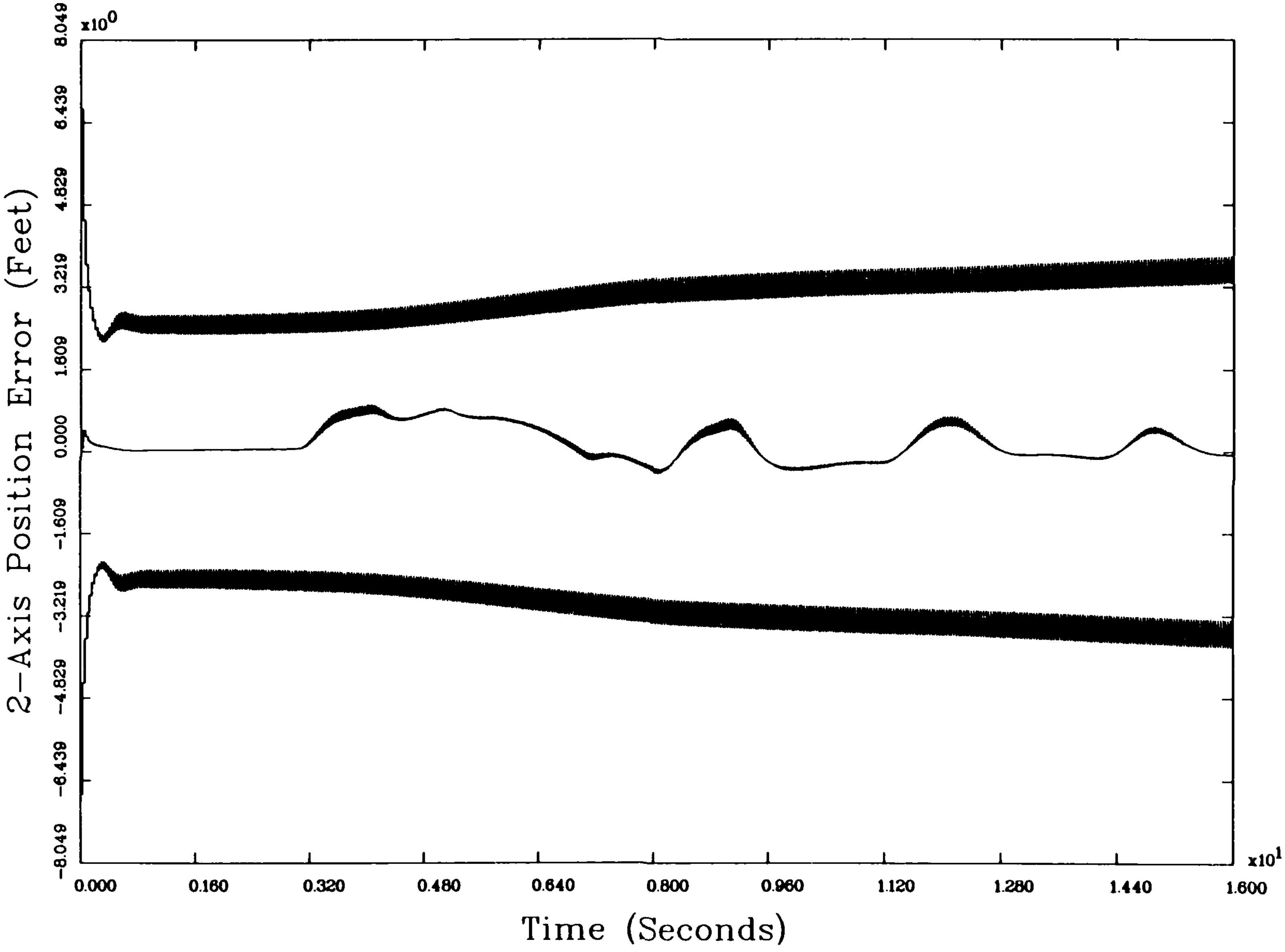


Figure 12.b. GM Filter, Trajectory 2, x_2

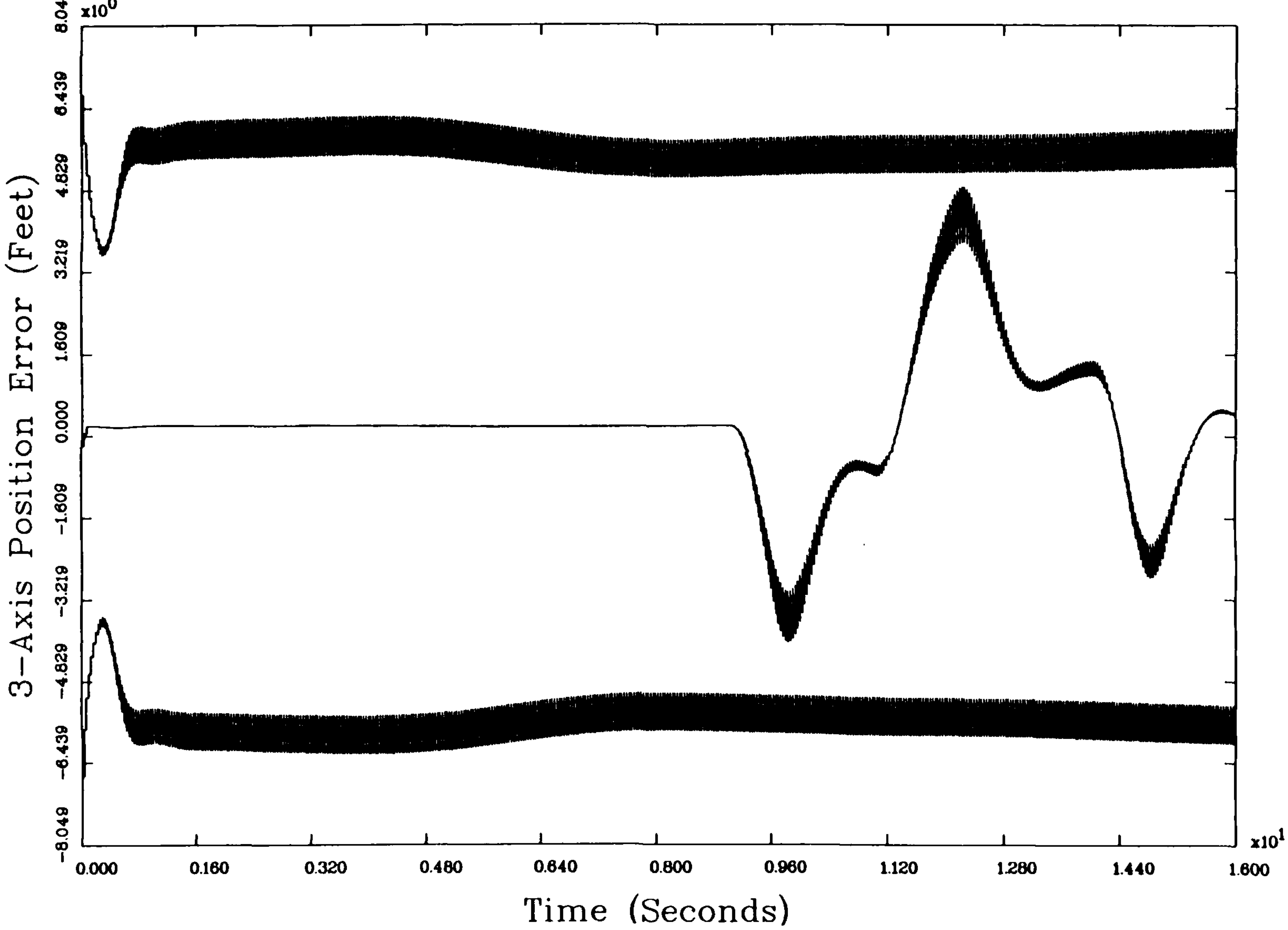


Figure 12.c. GM Filter, Trajectory 2, x_3

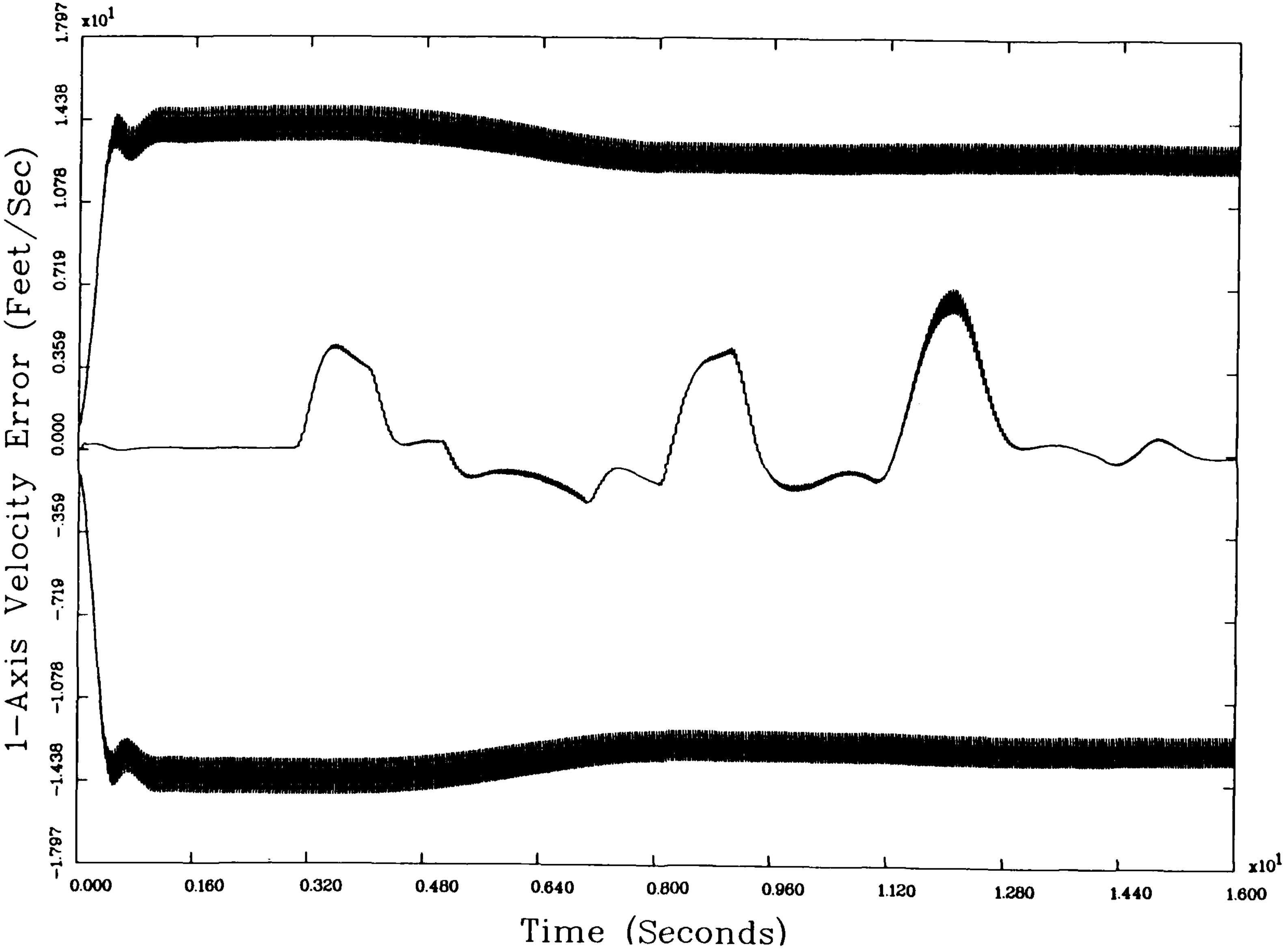


Figure 12.d. GM Filter, Trajectory 2, x_4

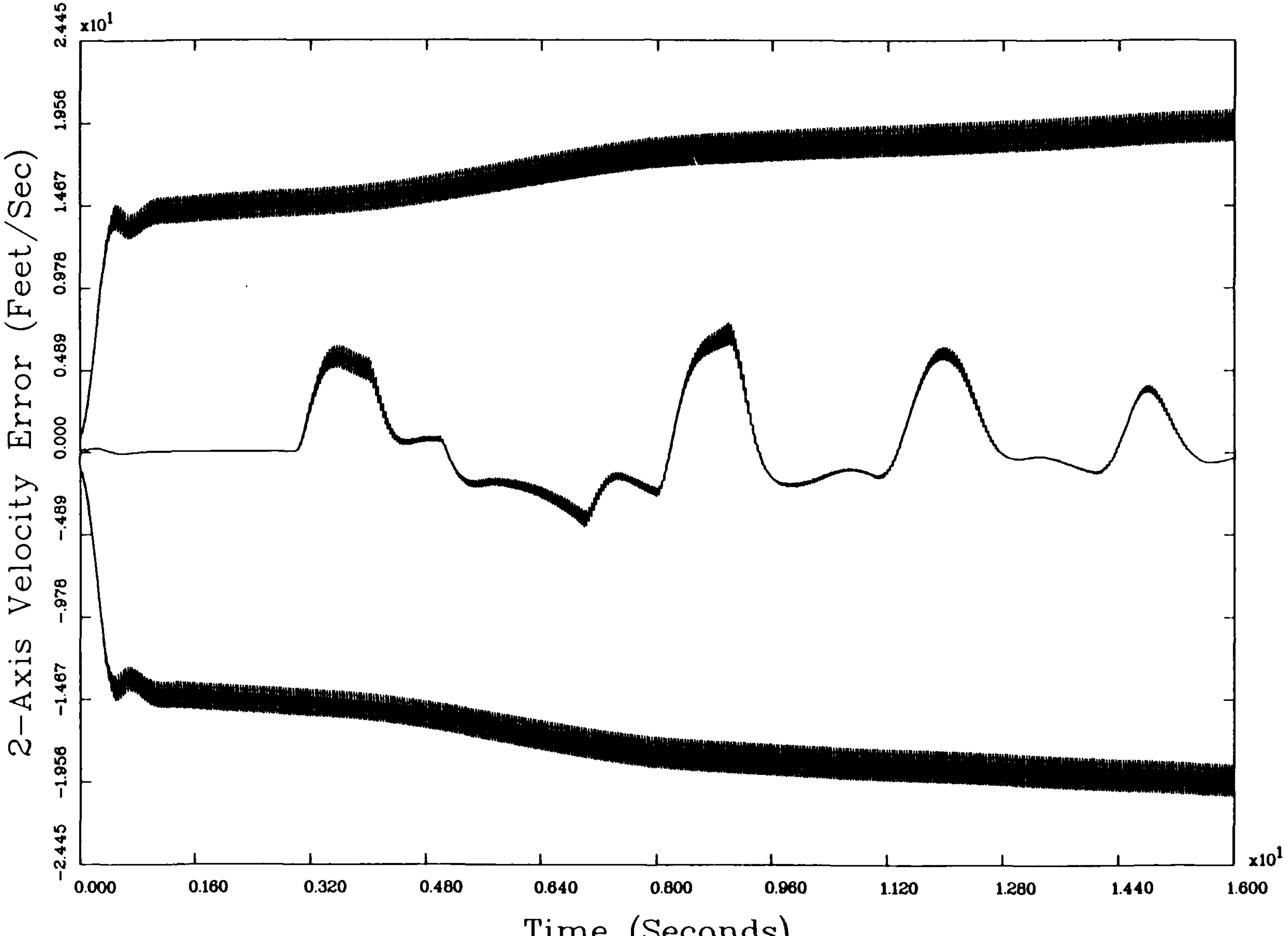


Figure 12.e. GM Filter, Trajectory 2, x_5

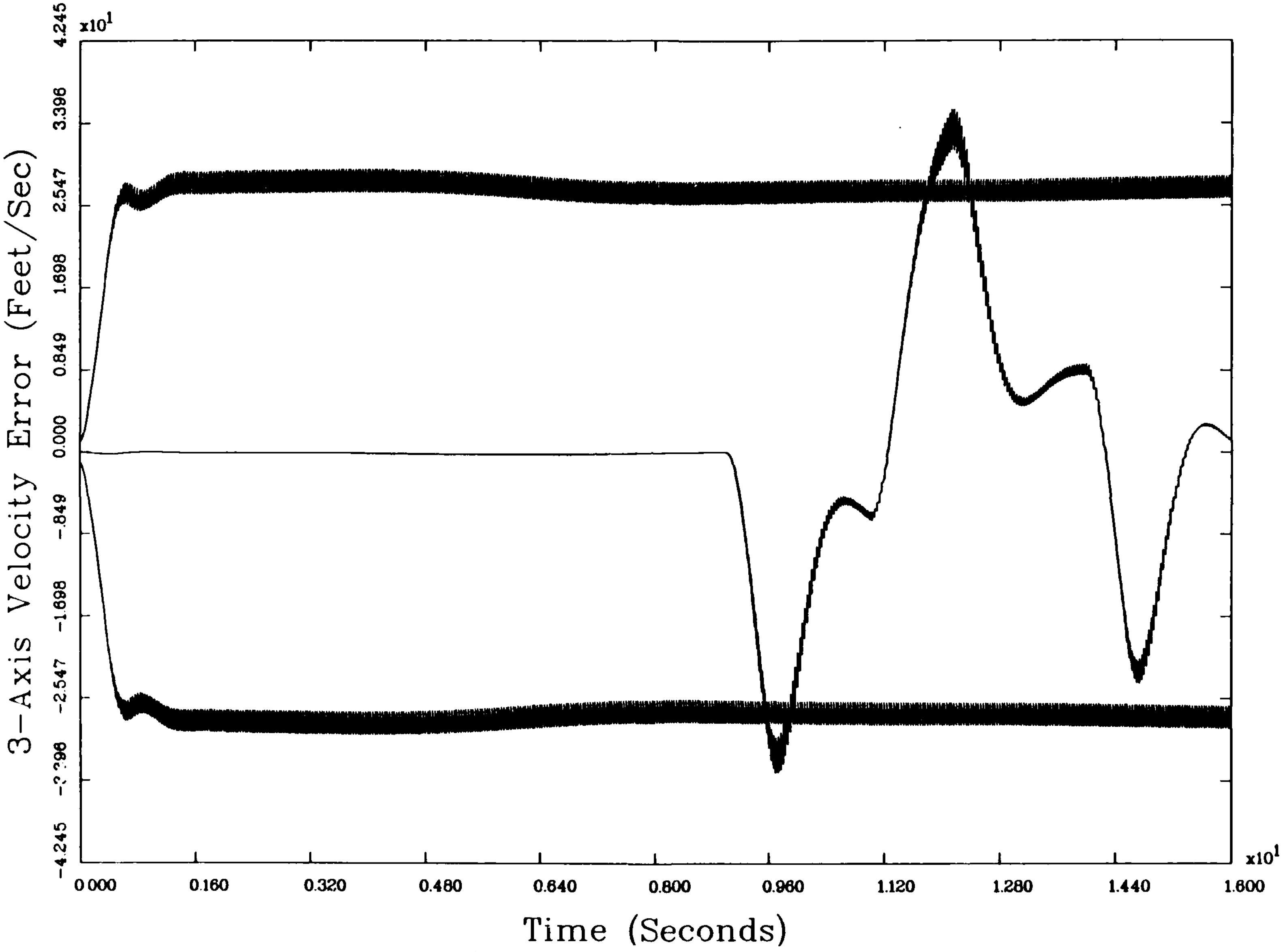


Figure 12.f. GM Filter, Trajectory 2, x_6

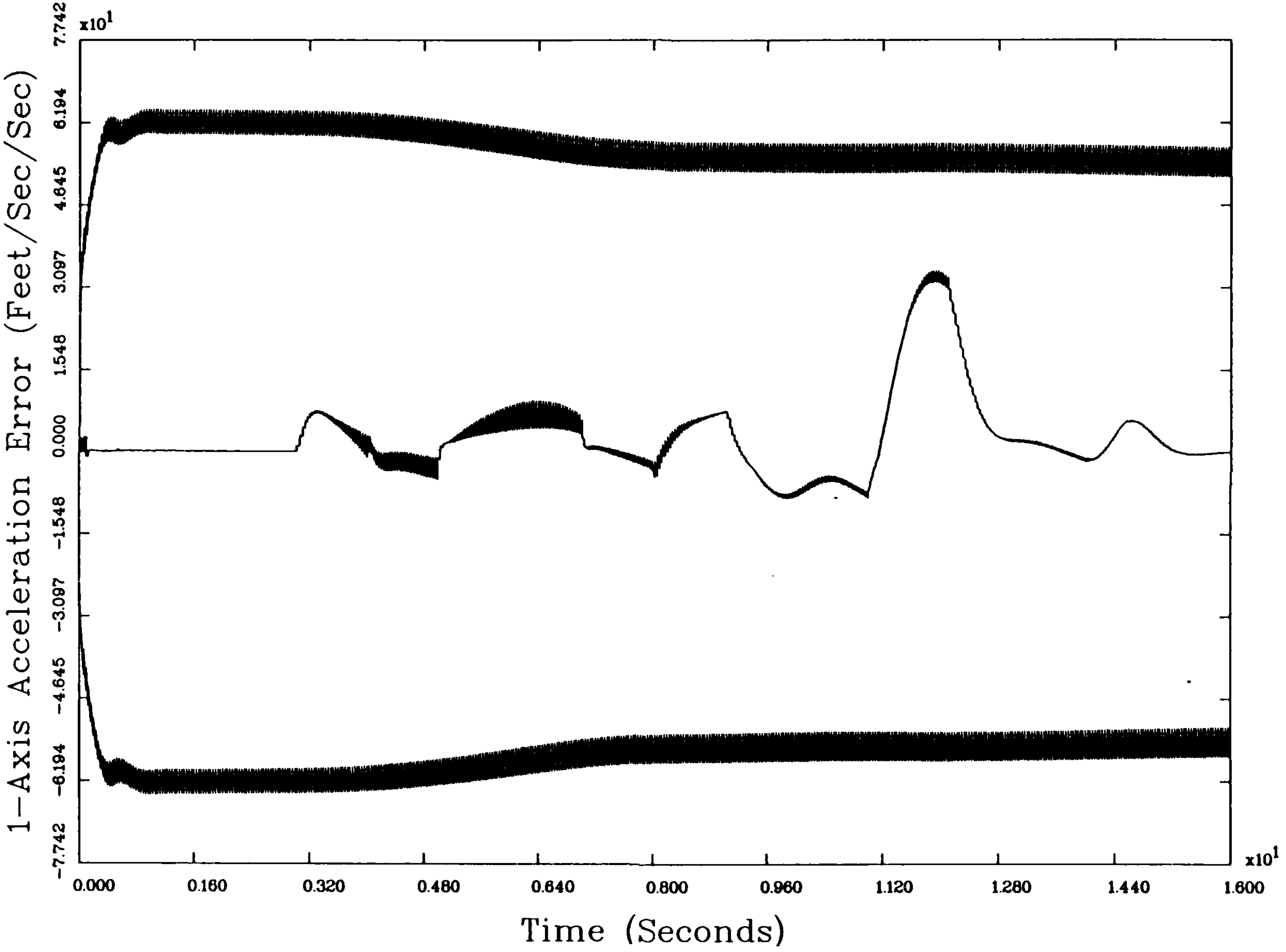


Figure 12.g. GM Filter, Trajectory 2, x_7

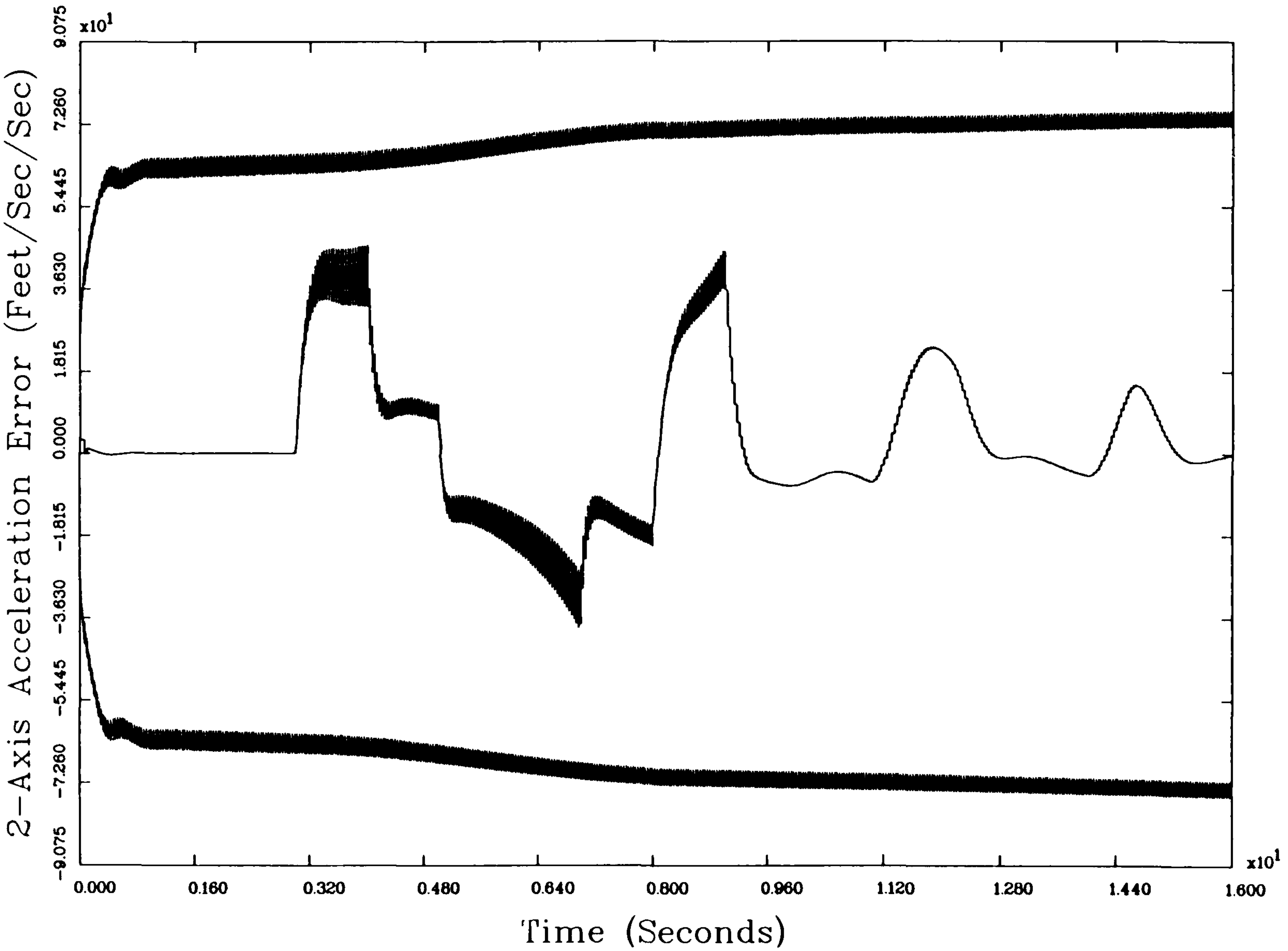


Figure 12.h. GM Filter, Trajectory 2, x_8

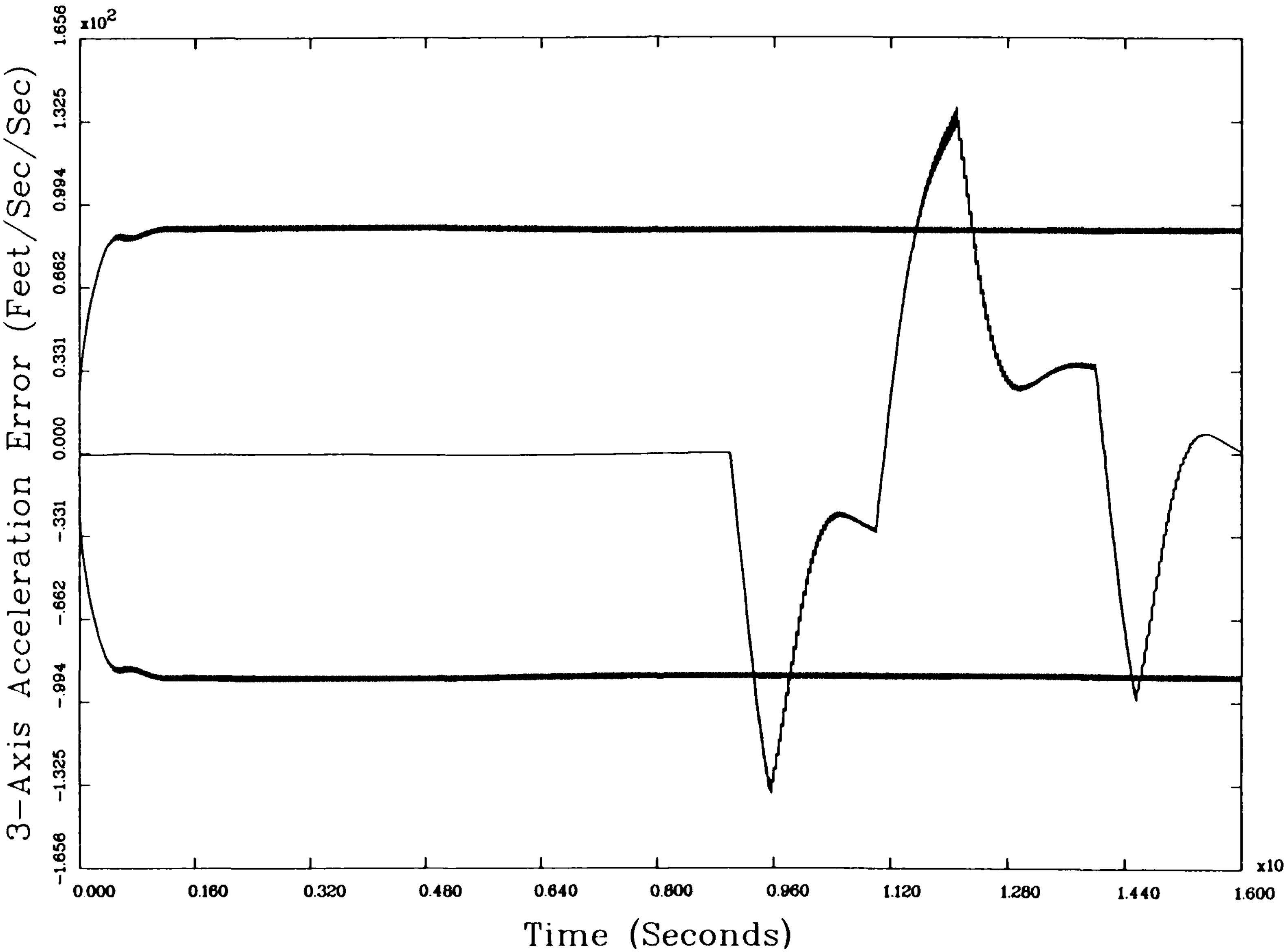


Figure 12.1. GM Filter, Trajectory 2, x_9

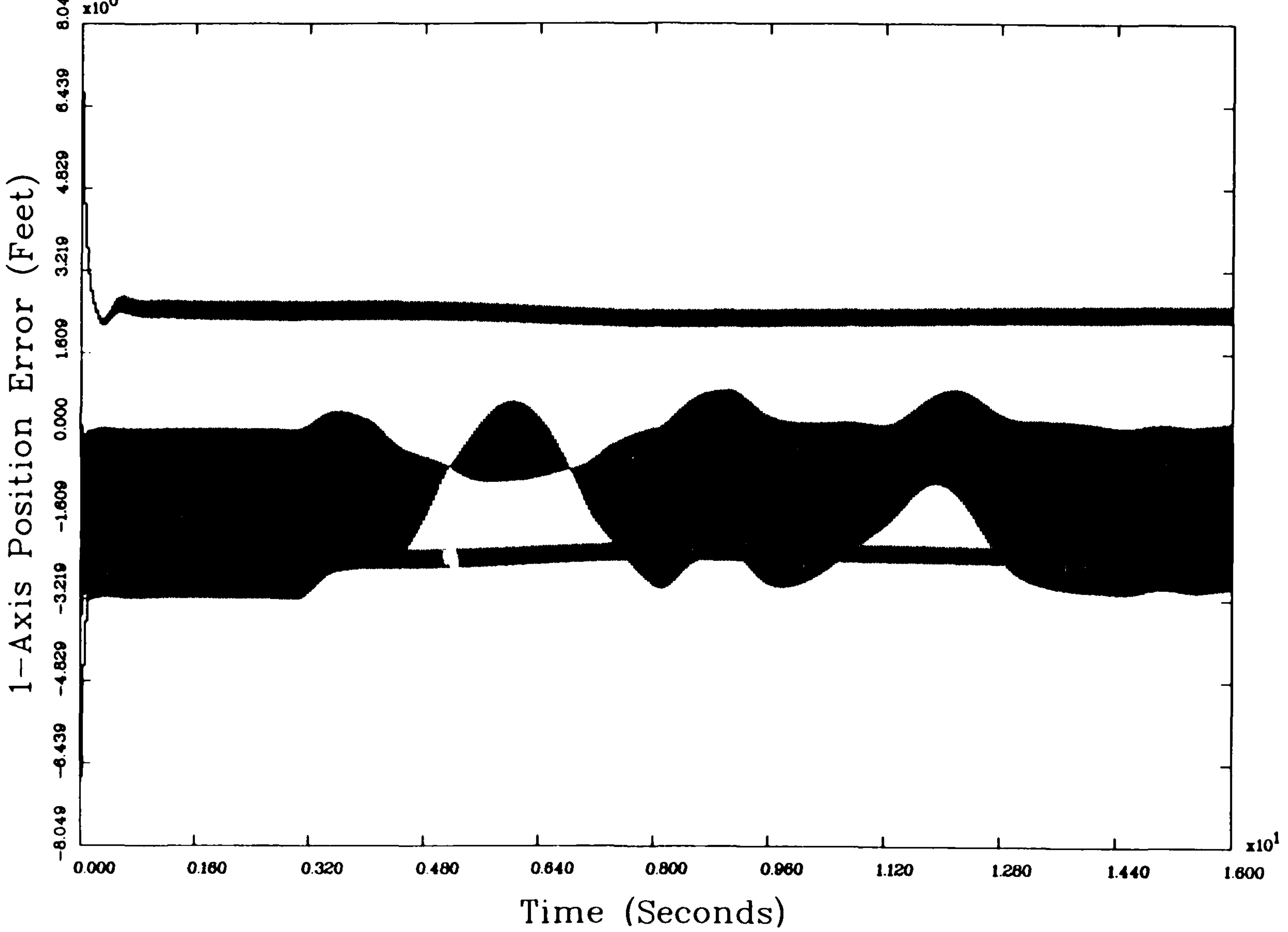


Figure 13.a. GM Filter, U-D Implementation, Trajectory 2, x_1

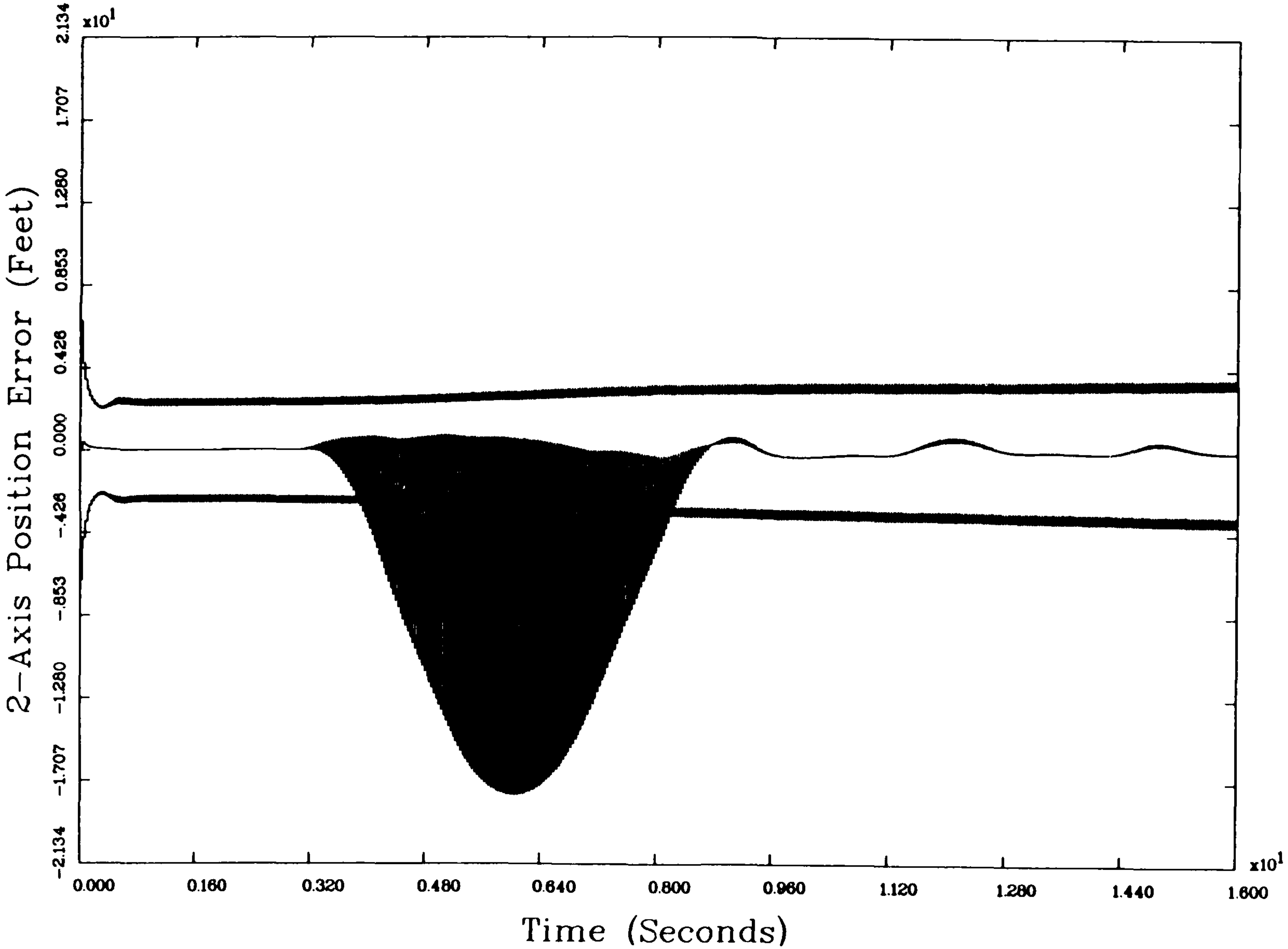


Figure 13.b. GM Filter, U-D Implementation, Trajectory 2, x_2

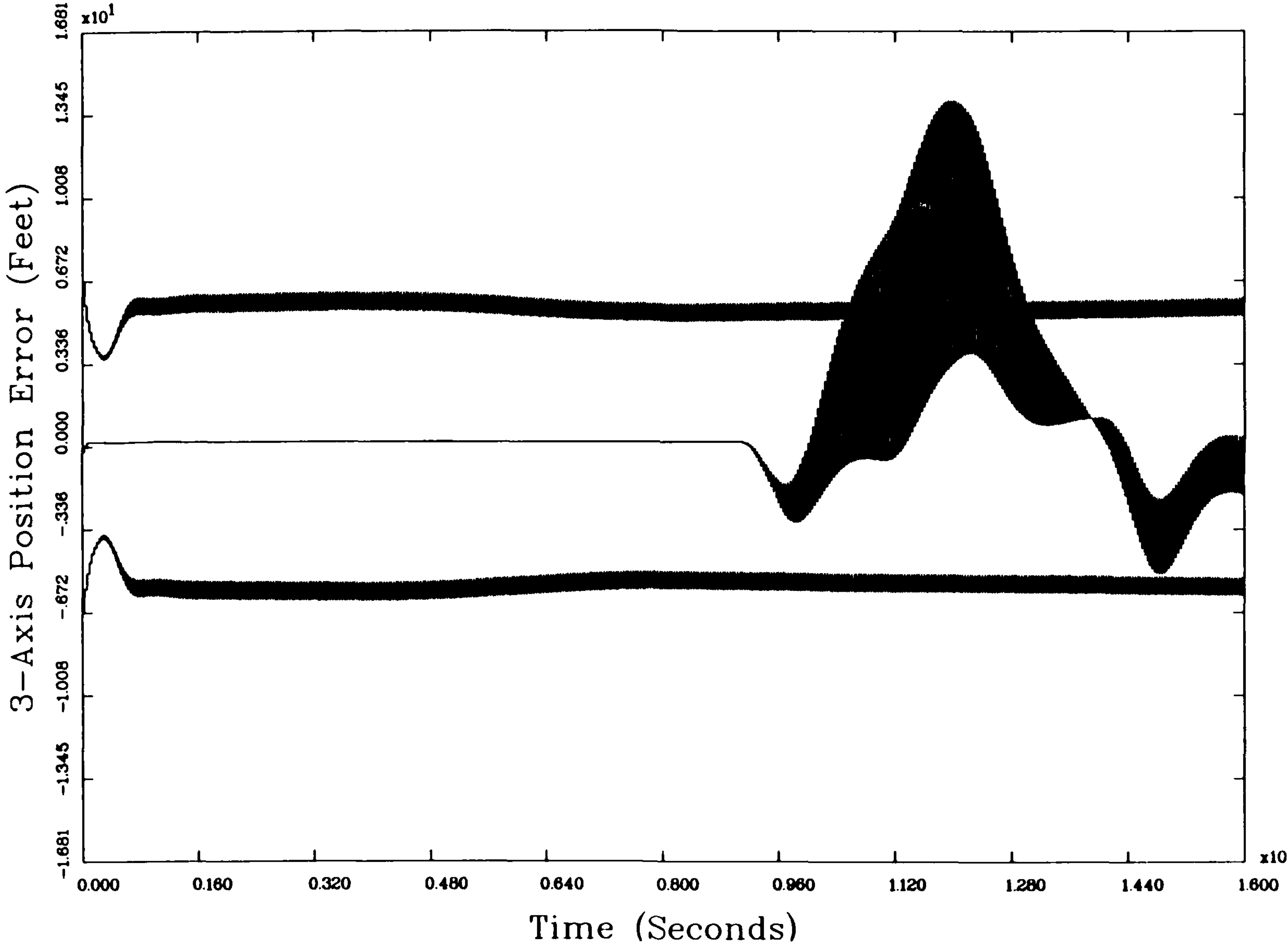


Figure 13.c. GM Filter, U-D Implementation, Trajectory 2, x_3

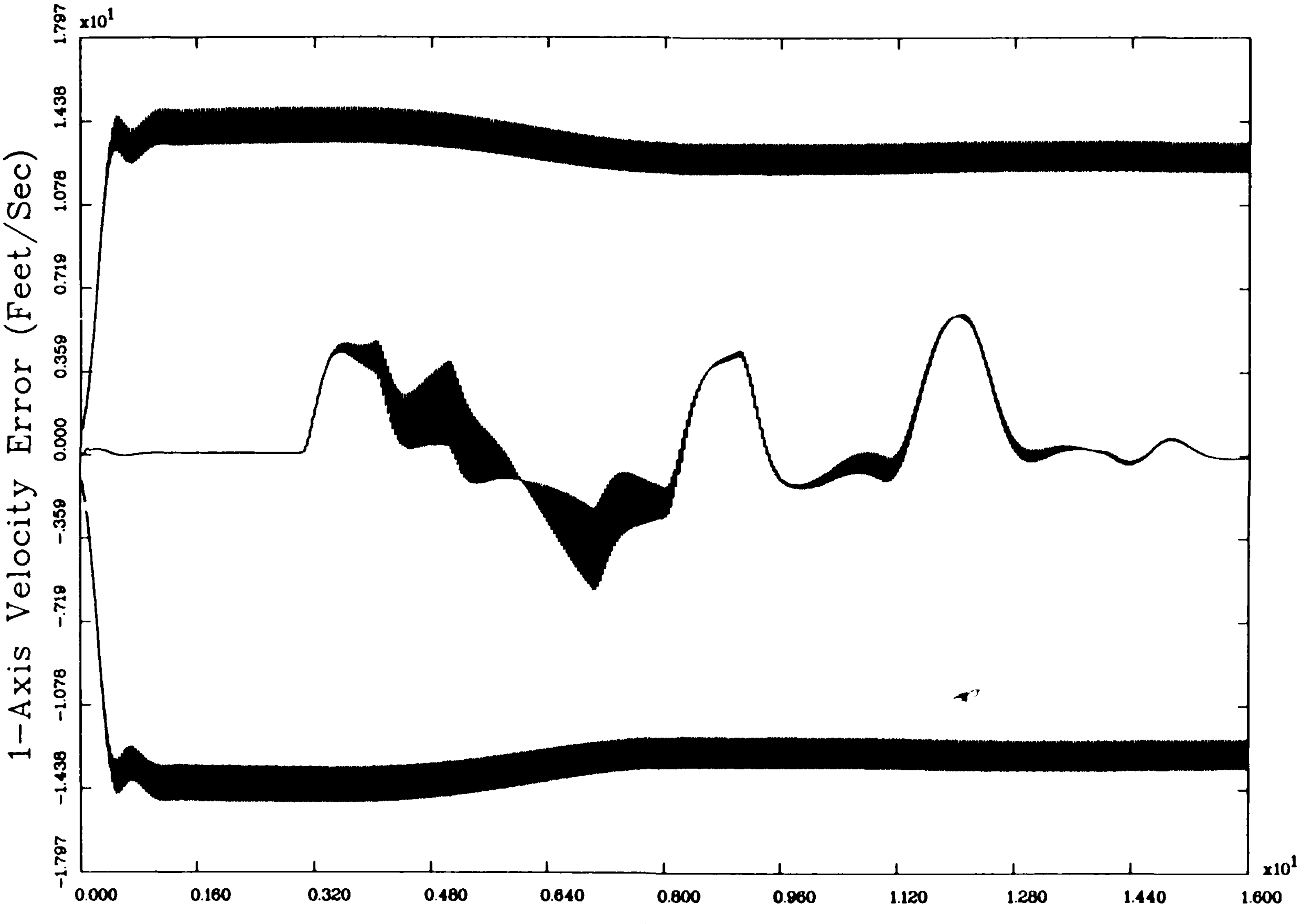


Figure 13.d. GM Filter, U-D Implementation, Trajectory 2, x_4

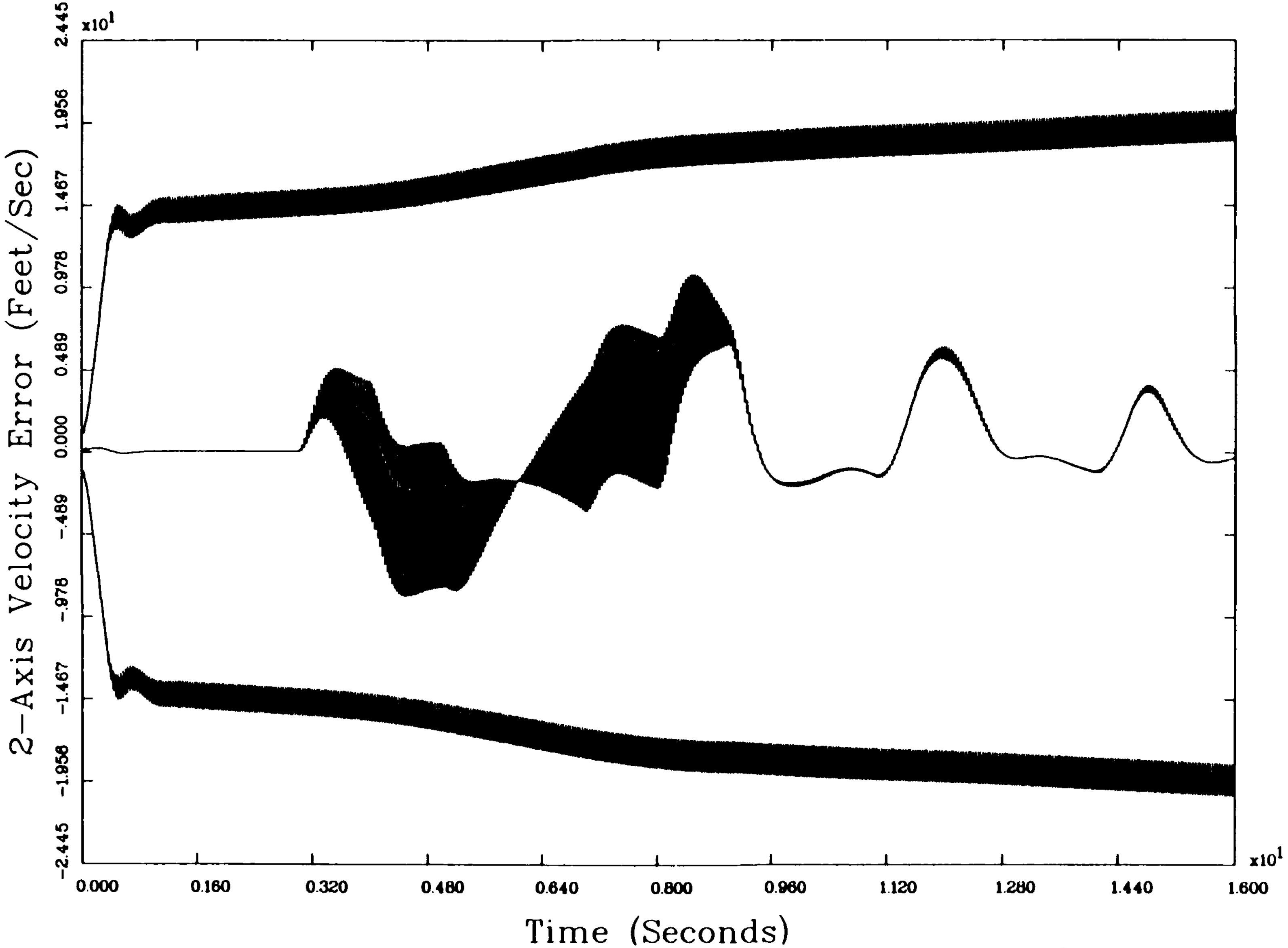


Figure 13.e. GM Filter, U-D Implementation, Trajectory 2, x_5

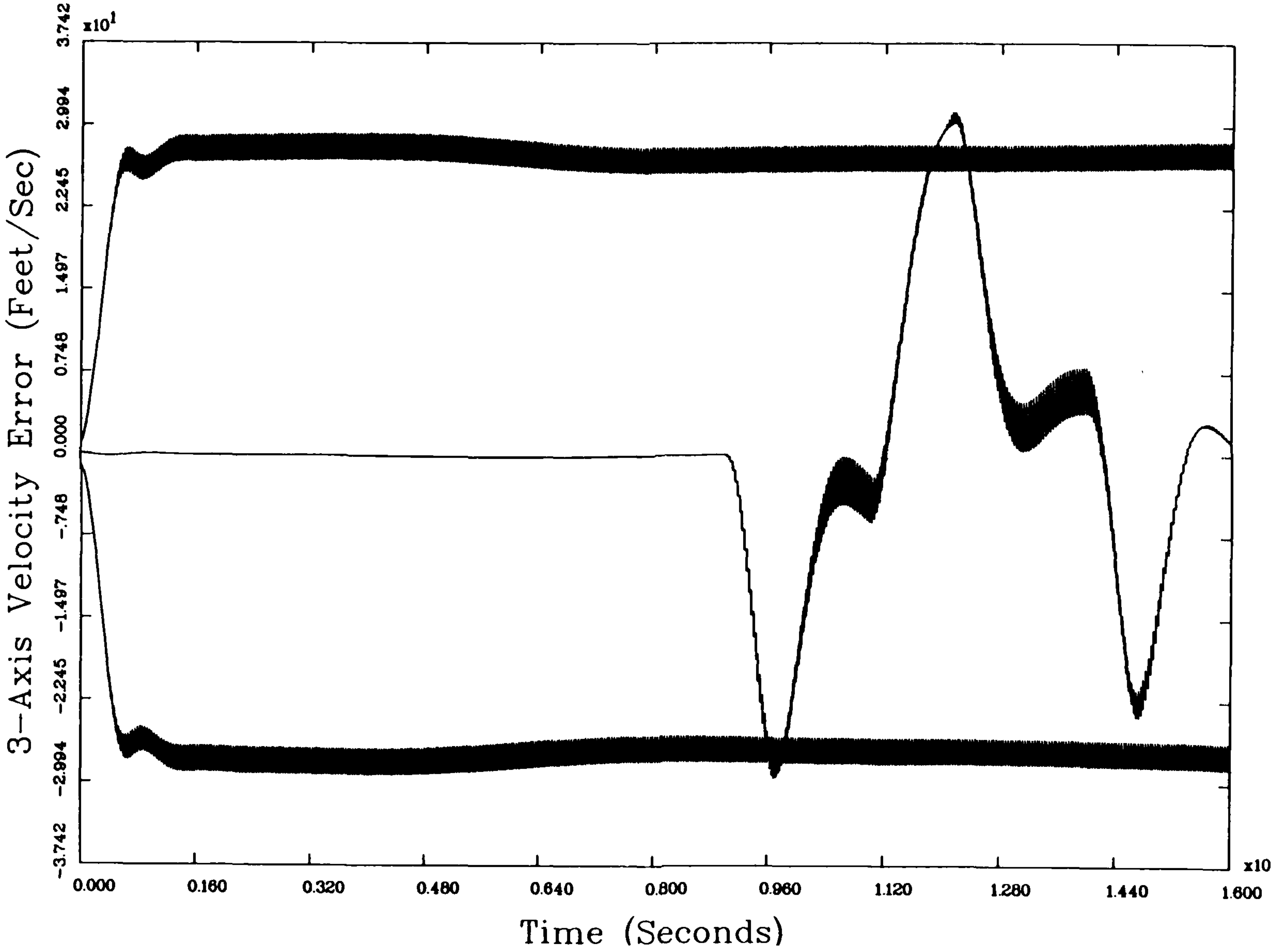


Figure 13.f. GM Filter, U-D Implementation, Trajectory 2, x_6

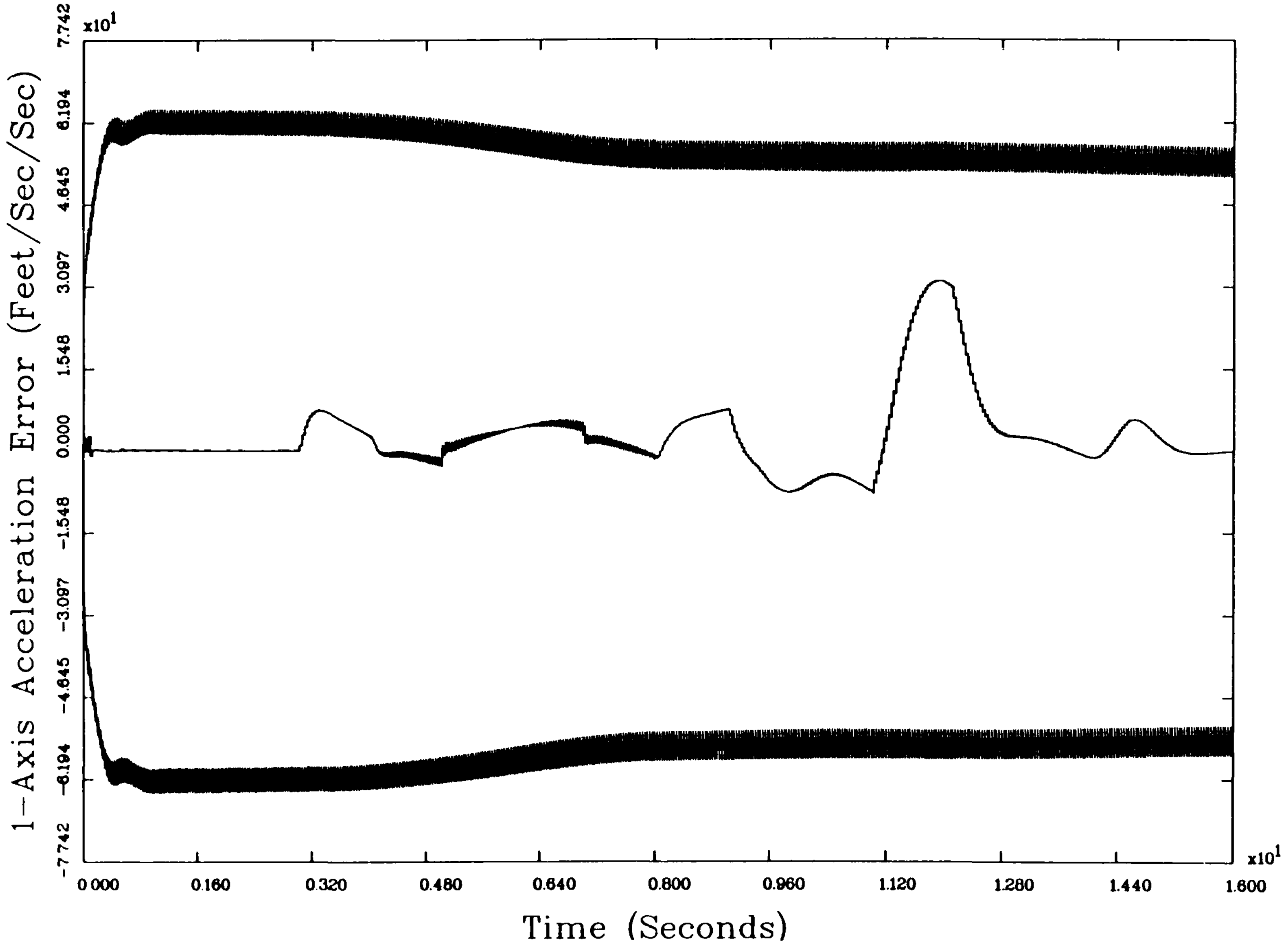


Figure 13.g. GM Filter, U-D Implementation, Trajectory 2, x_7

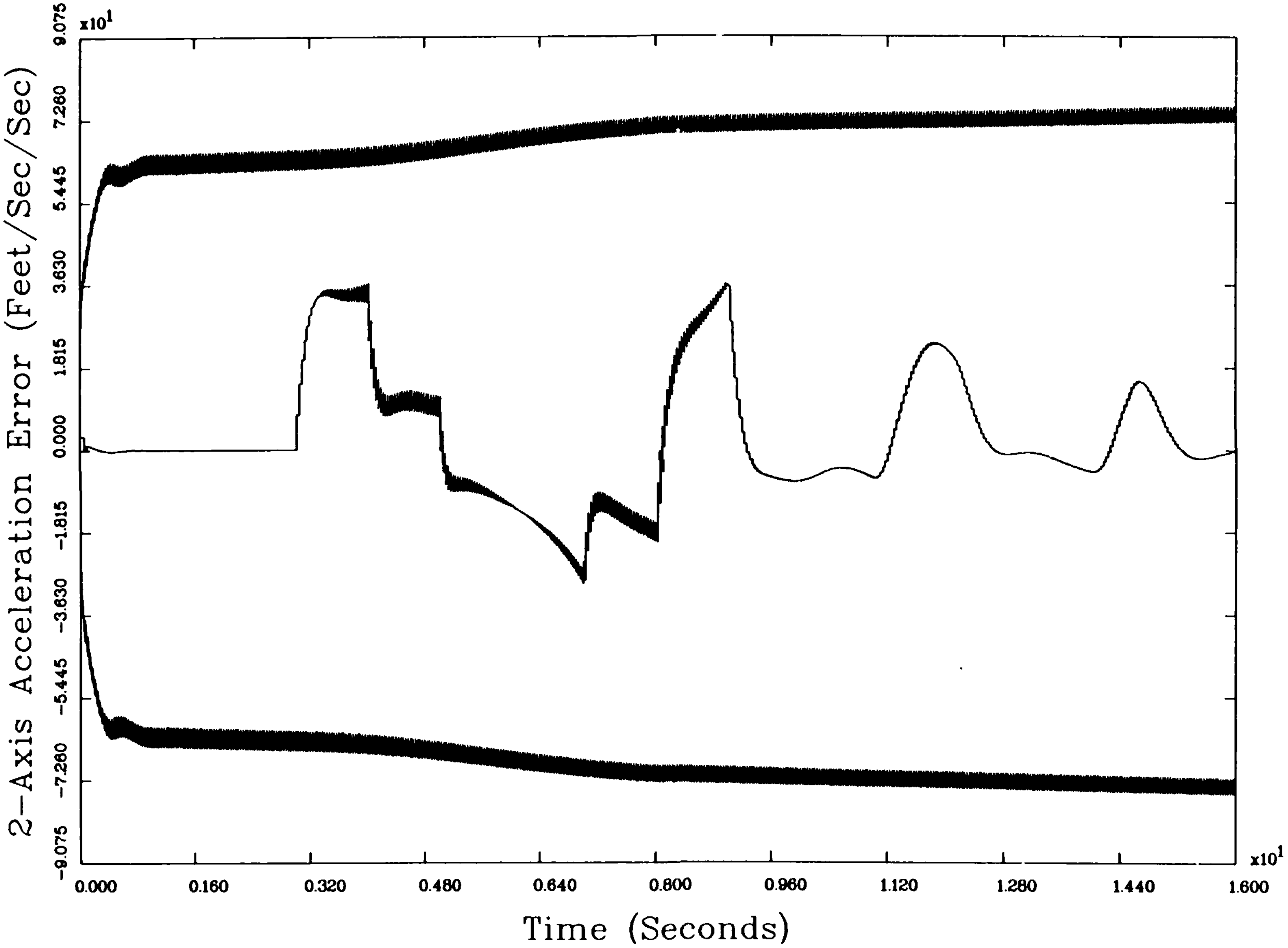


Figure 13.h. GM Filter, U-D Implementation, Trajectory 2, x_8

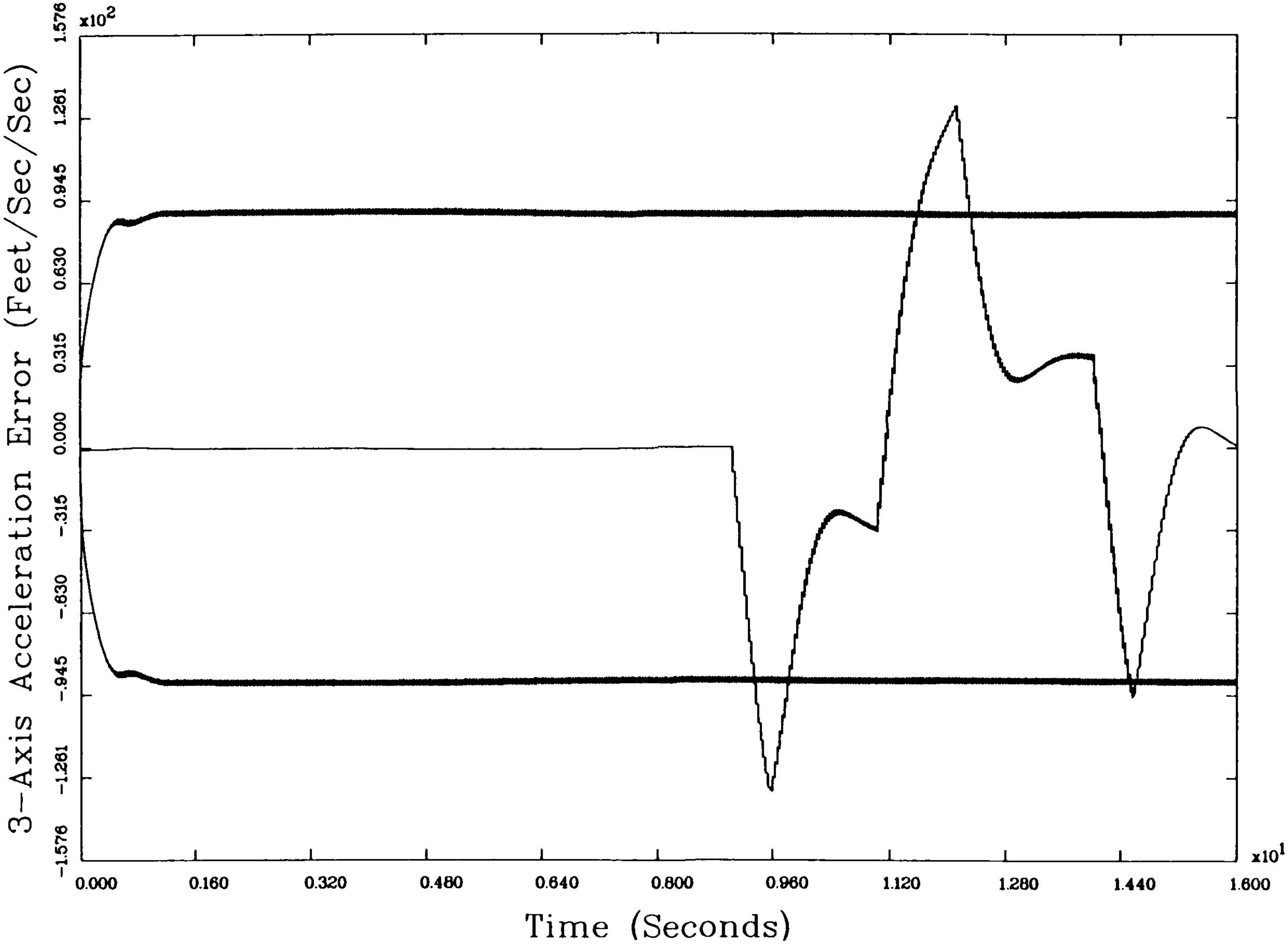


Figure 13.1. GM Filter, U-D Implementation, Trajectory 2 x_9

Bibliography

1. Bierman, Gerald J. "Measurement Updating Using the U-D Factorization." Proceedings of the 1975 IEEE Conference on Decision and Control, Houston, Tx (1975).

2. Barfield, Finley, AFTI/F-16 Automated Maneuvering and Attack System Project Engineer. Personal interviews. AFWAL/FIGX, Wright-Patterson AFB OH, February-May 1987.

3. Bryan, Ralph S. Cooperative Estimation of Targets by Multiple Aircraft. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1980 (ADA085799).

4. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 1. Academic Press, Inc., Orlando, 1979.

5. Maybeck, Peter S. Stochastic Models, Estimation, and Control, Volume 2. Academic Press, Inc., Orlando, 1982.

6. Musick, Stanton H. SOFE: A Generalized Digital Simulation for Optimal Filter Evaluation, User's Manual. Technical Report AFWAL-TR-80-1108, Avionics Laboratory, Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB OH, October 1980.

7. Musick, Stanton H. and others. SOFEPL: A Plotting Postprocessor for SOFE, User's Manual. Technical Report AFWAL-TR-80-1109, Avionics Laboratory, Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB OH, October 1980.

8. Worsley, William H. Comparison of Three Extended Kalman Filters for Air-to-Air Tracking. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1980 (ADA094767).

9. Worsley, William H., Air Force Institute of Technology (AFIT) Professor. Personal Interviews. AFIT/ENG, Wright-Patterson AFB, OH., February-December 1987.

10. -----. Data Obtained From AFWAL/ADPO, Wright-Patterson AFB, OH, February-December 1987.

VITA

Captain David L. Michalk was born on 12 November, 1957 in Woodland, California. He graduated from high school in Kettle Falls, Washington, in 1975 and attended New Mexico State University from 1976 to 1978. He joined the Air Force as an Airman Basic in June, 1979 and was assigned to duty at the 55th Avionics Maintenance Squadron, Offutt AFB, Nebraska. In 1981, he was selected for the Airman Education and Commissioning Program and returned to New Mexico State University. After graduation in May, 1983, he entered Officer Training School at Lackland AFB, Texas. Upon commissioning in August, 1983, he was assigned to the Air Force Electronic Warfare Center, Electronic Security Command, until entering the School of Engineering, Air Force Institute of Technology, in June, 1986.

Permanent address: 15025 Water Avenue
Visalia, California 93277

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/87D-44

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: School of Engineering

6b. OFFICE SYMBOL (If applicable): AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS (City, State, and ZIP Code):
Air Force Institue of Technology (AU)
Wright-Patterson AFB, Ohio, 45433-6583

7b. ADDRESS (City, State, and ZIP Code):

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Flight Dynamics Laboratory

8b. OFFICE SYMBOL (If applicable): AFWAL/FIGX

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS (City, State, and ZIP Code):
Air Force Wright Aeronautical Laboratories
Wright-Patterson AFB, Ohio, 45433-6583

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification): IMPLEMENTATION OF A TARGET STATE ESTIMATOR FOR THE AIR-TO-AIR ATTACK MODE OF THE AFTI/F-16

12. PERSONAL AUTHOR(S): David L. Michalk, B.S.E.E., Captain, USAF

13a. TYPE OF REPORT: MS Thesis

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT (Year, Month, Day): 1987 December

15. PAGE COUNT: 139

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
17	11	
20	03	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Target State Estimation, Kalman Filters, AFTI/F-16, Monte Carlo Method, Automatic Tracking

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advsor: William H. Worsley, Major, USAF
Professor of Electrical Engineering

Approved for public release: IAW AFR 190-1.
Lynn Wolaver 31 Dec 87
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL: William H. Worsley, Major, USAF

22b. TELEPHONE (Include Area Code): (513) 255-2024

22c. OFFICE SYMBOL: AFIT/ENG

DD Form 1473, JUN 86

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

BLOCK 19 Continued

The purpose of this study was to investigate target state estimation techniques for the air-to-air mode of the AFTI/F-16 automated maneuvering and attack system. The target state estimator (TSE) previously developed would not perform to specifications; possible reasons for this poor performance are presented as well as suggestions to upgrade the performance.

This study focused on three areas: (1) Determine if the Gauss-Markov dynamics model used in the current TSE was adequate for the tracking accuracies specified; (2) Determine if a rotation had to be performed to account for the states being expressed in one frame while the measurements were physically made in another. (3) Determine what effect the conventional covariance updates, coupled with the short (16-bit) wordlength of the TSE computers, has on the stability of the Kalman filter.

Two filter dynamics models were designed, tested, and compared. The first model used complex equations and closely modeled an air-to-air engagement. Most of the complexity of the model was maintained in its implementation, and it was used as a baseline model. The second filter used a Gauss-Markov dynamics model and made several assumptions to simplify computations.

Analysis of filter performance revealed that the Gauss-Markov filter dynamics model was, indeed, an adequate model. Also, the covariance matrix does not have to be rotated into the LOS frame if the measurements are redefined. The second filter was then implemented using U-D covariance factorization algorithms, but the time propagation routines used were apparently flawed. However, the poor performance of the TSE is no doubt caused by the conventional Kalman filter recursions, as they are inherently unstable.

END
DATE
FILMED
MARCH
1988
DTIC